

AD-A058 193

DAYTON UNIV OHIO RESEARCH INST

F/G 9/2

AFML SCIENTIFIC AND ENGINEERING COMPUTER SUPPORT.(U)

DEC 77 D L FORD, T WOOD, M DENNIS, D G LEET

F33615-75-C-5191

UNCLASSIFIED

UDRI-TR-77-10

AFML-TR-77-231

NL

1 OF 2
AD
A058193



ADA058193

AFML-TR-77-231 ✓

AFML SCIENTIFIC AND ENGINEERING COMPUTER SUPPORT

UNIVERSITY OF DAYTON
DAYTON, OHIO

DECEMBER 1977

FINAL REPORT

15 MAY 1975 - NOVEMBER 1976

AD No. _____
DDC FILE COPY

APPROVED FOR PUBLIC RELEASE; DISTRIBUTION UNLIMITED

~~THIS DOCUMENT IS BEST QUALITY PRACTICABLE.~~
~~THE COPY FURNISHED TO DDC CONTAINED A~~
~~SIGNIFICANT NUMBER OF PAGES WHICH DO NOT~~
~~REPRODUCE LEGIBLY.~~

AIR FORCE MATERIALS LABORATORY
AIR FORCE WRIGHT AERONAUTICAL LABORATORIES
AIR FORCE SYSTEMS COMMAND
WRIGHT-PATTERSON AIR FORCE BASE, OHIO 45433



78 06 15 018

78 17 08 094

2

LEVEL II

NOTICE


When Government drawings, specifications, or other data are used for any purpose other than in connection with a definitely related Government procurement operation, the United States Government thereby incurs no responsibility nor any obligation whatsoever, and the fact that the government may have formulated, furnished, or in any way supplied the said drawings, specifications, or other data, is not to be regarded by implication or otherwise as in any manner licensing the holder or any other person or corporation, or conveying any rights or permission to manufacture, use, or sell any patented invention that may in any way be related thereto.

This report has been reviewed by the Information Office (IO) and is releasable to the National Technical Information Service (NTIS). At NTIS, it will be available to the general public, including foreign nations.

This technical report has been reviewed and is approved for publication.


ALAN R. MILLER, CAPT, USAF
Project Monitor

FOR THE DIRECTOR


WARREN P. JOHNSON
Chief, Operations Office

"If your address has changed, if you wish to be removed from our mailing list, or if the addressee is no longer employed by your organization please notify AFML/DOC, WPAFB, OH 45433 to help us maintain a current mailing list".

Copies of this report should not be returned unless return is required by security considerations, contractual obligations, or notice on a specific document.

X

DISCLAIMER NOTICE

**THIS DOCUMENT IS BEST QUALITY
PRACTICABLE. THE COPY FURNISHED
TO DDC CONTAINED A SIGNIFICANT
NUMBER OF PAGES WHICH DO NOT
REPRODUCE LEGIBLY.**

(18) (19) REPORT DOCUMENTATION PAGE		READ INSTRUCTIONS BEFORE COMPLETING FORM	
1. REPORT NUMBER AFML-TR-77-231		2. GOVT ACCESSION NO. 3. RECIPIENT'S CATALOG NUMBER	
4. TITLE (and Subtitle) AFML SCIENTIFIC AND ENGINEERING COMPUTER SUPPORT		5. TYPE OF REPORT & PERIOD COVERED Final Report 15 May 75-15 November 76	
7. AUTHOR(s) Dale L. Ford, Thomas Wood, Michael Dennis Duane G. Leet		6. PERFORMING ORG. REPORT NUMBER UDRI-TR-77-18	
9. PERFORMING ORGANIZATION NAME AND ADDRESS University of Dayton Research Institute 300 College Park Avenue Dayton, Ohio 45469		8. CONTRACT OR GRANT NUMBER(s) F33615-75-C-5191	
11. CONTROLLING OFFICE NAME AND ADDRESS Air Force Materials Laboratory (DOC) Air Force Systems Command Wright-Patterson AFB, Ohio 45433		10. PROGRAM ELEMENT, PROJECT, TASK AREA & WORK UNIT NUMBERS Project Nr: 6106 Task Nr: 99930000 Work Unit Nr: 99930000	
14. MONITORING AGENCY NAME & ADDRESS (if different from Controlling Office) F33615-75-C-5191		12. REPORT DATE December 1977	
		13. NUMBER OF PAGES 187	
		15. SECURITY CLASS. (of this report) UNCLASSIFIED	
		15a. DECLASSIFICATION/DOWNGRADING SCHEDULE	
16. DISTRIBUTION STATEMENT (of this Report) Approved for Public Release; Distribution Unlimited 16 6106 9993 17 00			
17. DISTRIBUTION STATEMENT (of the abstract entered in Block 20, if different from Report) 12 189 P.			
18. SUPPLEMENTARY NOTES			
19. KEY WORDS (Continue on reverse side if necessary and identify by block number)			
20. ABSTRACT (Continue on reverse side if necessary and identify by block number) This report presents a summary of the work performed by the University of Dayton Research Institute under Contract No. F33615-75-C-5191 during 15 May 1975 to 15 November 1976. The work was accomplished under the Computer Activities Office. Air Force Materials Laboratory, Wright-Patterson Air Force Base, Ohio			

FOREWORD

This final report was prepared by the University of Dayton Research Institute, Dayton, Ohio, under United States Air Force Contract F33615-75-C-5191. This contract was initiated under Project No. 6106. The work was administrated under the direction Computer Activities Office, Air Force Materials Laboratory, Wright-Patterson Air Force Base, Ohio.

This report covers scientific and engineering computer support supplied from 15 May 1975 to 15 November 1976. ✓

Personnel who have contributed to this work are: Dale L. Ford, Thomas Wood, Michael Dennis, and Duane G. Leet.

ACCESSION for	
NTIS	White Section <input checked="" type="checkbox"/>
DDC	Buff Section <input type="checkbox"/>
UNANNOUNCED	<input type="checkbox"/>
JUSTIFICATION.....	
BY.....	
DISTRIBUTION/AVAILABILITY CODES	
Dist.	AVAIL. and/or SPECIAL
A 23	

78 06 15 018

TABLE OF CONTENTS

Section	Page
I. INTRODUCTION	1
II. AUTOMATED EXPERIMENTS AND DATA ACQUISITION	2
2.1 AFML DIGITAL COMPUTER HIERARCHY	2
2.2 SELECTION OF A LABORATORY STANDARD MICRO-COMPUTER	4
2.3 SPECIFIC APPLICATIONS	5
2.3.1 Liquid Adsorption Experiment Automation	5
2.3.2 Polymer-Surface Interaction Experiment Automation	9
2.3.3 Cut-Bar Acquisition System	12
2.3.4 Data Acquisition of Lubricants and Slip Ring Experiment	12
2.3.5 Data Acquisition System for MTS Materials Testing Rigs	16
2.3.6 Automation of a Stress Intensity Factor Calibration Experiment	18
2.3.7 Support of the LSI-11 Software Development System	18
2.3.8 Data Acquisition System for the LHML Facility	20
III. COMPUTER PROGRAMS FOR RE-ENTRY AND LASER APPLICATIONS OF ABLATION/THERMAL EFFECTS	23
3.1 INTRODUCTION	23
3.2 APPLICATIONS OF THE APPLE CODE	24
3.2.1 Introduction	24
3.2.2 Thermal Response of "Painted" Aluminum to Radiant Heating	27
3.2.3 Thermal and Stress Analyses of the AFWL Zinc Selenide Laser Window	28
3.2.4 Computed Temperature Response of the AFML Flat-Top Laser Window	29
3.3 UNIFIED COMPUTER PROGRAMS	29
3.3.1 Introduction	29
3.3.2 Sensitivity Study Using CMA	30
IV. COMPUTER ANALYSIS OF MICROSTRUCTURES	36
4.1 COMPUTER RECOGNITION OF TITANIUM ALLOY MICRO-STRUCTURES	36
4.1.1 Introduction	36
4.1.2 Initial Work	36
4.1.3 Improvements of the Digital Processing	37
4.1.4 Improvement of Image Thresholding	38

TABLE OF CONTENTS (Continued)

Section	Page
4.1.5 Improvements in Particle Separation/ Extraction	39
4.1.6 Development of New Particle Separation/ Extraction Software	39
4.1.7 Additional Software Developed for Testing and Evaluation Purposes	42
4.2 COMPUTER MODELING OF CRACK PROPAGATION IN THE SURFACE OF TI-ALLOYS	43
4.2.1 Problem Description	43
4.2.2 System Requirements	43
4.2.3 Computer Representation of the Alloy Surface	44
4.2.4 Linkage Table Generation	45
4.2.5 Crack Propagation	45
4.2.6 Additional Capabilities	46
V. GENERAL USER PROGRAMS AND EDUCATION	47
5.1 INTRODUCTION	47
5.1.1 General User Programs for the CDC 6600	47
5.1.2 LSI-11 Peripheral Driver Programs	48
5.1.3 PLM Support Programs	48
5.2 AFML USER EDUCATION	50
5.2.1 CDC 6600 Users	50
5.2.2 LSI-11 Users	51
APPENDIX A: LIQUID ADSORPTION EXPERIMENT PLM PROGRAM	53
APPENDIX B: CUT-BAR EXPERIMENT PLM PROGRAM	65
APPENDIX C: DEVELOPMENT SYSTEM PDP-11 ASSEMBLY LANGUAGE PROGRAM	69
APPENDIX D: USERS MANUAL FOR THE DATA ACQUISITION SYSTEM IN THE LHMEL FACILITY	72
APPENDIX E: FORTRAN PROGRAM SER TO CALCULATE RESPONSE FUNCTIONS	99
APPENDIX F: FORTRAN PROGRAMS FOR ANALYSIS OF MICROSTRUC- TURES	103
APPENDIX G: DIGIPLT USERS GUIDE	141
APPENDIX H: GENERAL USER PROGRAMS FOR THE CDC 6600	167
APPENDIX I: LSI-11 PERIPHERAL DRIVER PROGRAMS	175
APPENDIX J: PLM SUPPORT PROGRAMS FOR THE CONTROL LOGIC MICROPROCESSORS	179
REFERENCES	184

LIST OF FIGURES

Figure		Page
1	AFML Digital Computer Hierarchy.	3
2	Surface Analysis Experiment Block Diagram.	6
3	Control System for Surface Analysis Experiment.	7
4	Software Flowchart.	8
5	Automation of Polymer-Surface Interaction Experiment	10
6	HP 2100 to Mass Spectrometer Interface	11
7	Block Diagram of the Cut-Bar Experiment	13
8	Lubricants Data Acquisition System Block Diagram	15
9	Data Acquisition System for the MTS Testing Rigs	17
10	Block Diagram of the Stress Intensity Factor Experiment Automation	19

LIST OF TABLES

Table		Page
1	Thermophysical Properties of Carbon Phenolic	31
2	Thermophysical Properties That Were Varied +10% Each, Separately	32
3	Material Response Parameters (R_j)	33
4	Sensitivity Coefficients for the Interface Temperature	34
5	Calling Assembly Language Subroutines from RT-11 FORTRAN	49

SECTION I INTRODUCTION

The application of computers in research programs is rapidly increasing due to the inherent advantages possible in the use of these machines. Experiments that were impractical or impossible with human operators are now feasible, mathematical models can be constructed and readily evaluated, sophisticated data reduction involving millions of calculations can be performed in seconds to enhance data analysis, computer graphics are providing an interactive dialogue with the computer to evaluate model building. All of these capabilities are being exploited to a greater and greater degree as one's awareness of the computer's capability increases and the relative time and cost per calculation continues to decrease. The Air Force Materials Laboratory use of computer has also been increasing. While the majority of the past computer experience of the Materials Laboratory has been with batch jobs to a large computer such as the CDC 6600 and high cost, specifically designed minicomputer systems, the introduction of the microprocessor has provided a flexible, low cost solution to obtaining maximum efficiency in in-house and on-site research programs.

↘ This report summarizes the work performed in providing scientific and engineering computer support to various projects within AFML during this time period. This work includes exploitation of all computers available to AFML researchers. While the results of this work have been or are in the process of being published in either technical reports or scientific journals, completion of some projects and publication of the results will occur on a succeeding contract.

→ This report is divided into major topics: (1) Automated Experiments and Data Acquisition, (2) Computer Programs for Re-entry and Laser Applications of Ablation/Thermal Effects, (3) Computer Analysis of Microstructures, and (4) General User Programs and Education. ↗

SECTION II

AUTOMATED EXPERIMENTS AND DATA ACQUISITION

2.1 AFML DIGITAL COMPUTER HIERARCHY

It is the object of the Computer Activities Office of the AF Materials Laboratory to organize a digital computer hierarchy so that with the automation of laboratory experiments, data from the experiments can be transmitted to other digital computers within ASD for data reduction. A block diagram of this hierarchy is shown in Figure 1. This communications requires the use of telephone communications equipment such as modems, acoustic couplers, audio telephone lines, and hard-wired telephone lines. For the Materials Laboratory, the central point of this hierarchy is the SEL 86 located in Building 652. Data from the experiments is gathered by the mini and microcomputers and transmitted to the SEL via either low speed (110, 300 Baud) or high speed (9600 Baud) telephone lines.

The low speed links will communicate with SEL 86 through a program in the SEL called TSS (Terminal Support System). Thus the minicomputers must act as terminals when communicating with the SEL 86. The high speed links will communicate with the SEL 86 through a high speed multiplexer called a MAX. A high speed link is a dedicated link and is active at all times, so no "login" is necessary, but a great deal of data formulating is required. The SEL 86 driver for the high speed links allows such features as downstream loading, mass data storage for the experiment on the SEL, and transmission of both binary and ASCII data.

The mini and microcomputers in the laboratory consist of the following makes and models: HP 2100 (paper tape, cassette, mag tape, and disk operating systems), PDP-11 paper tape and floppy disk system), PDP-8 (disk system), Varian 620/L (disk system), and Control Logic (Intel 8080) (paper tape system).

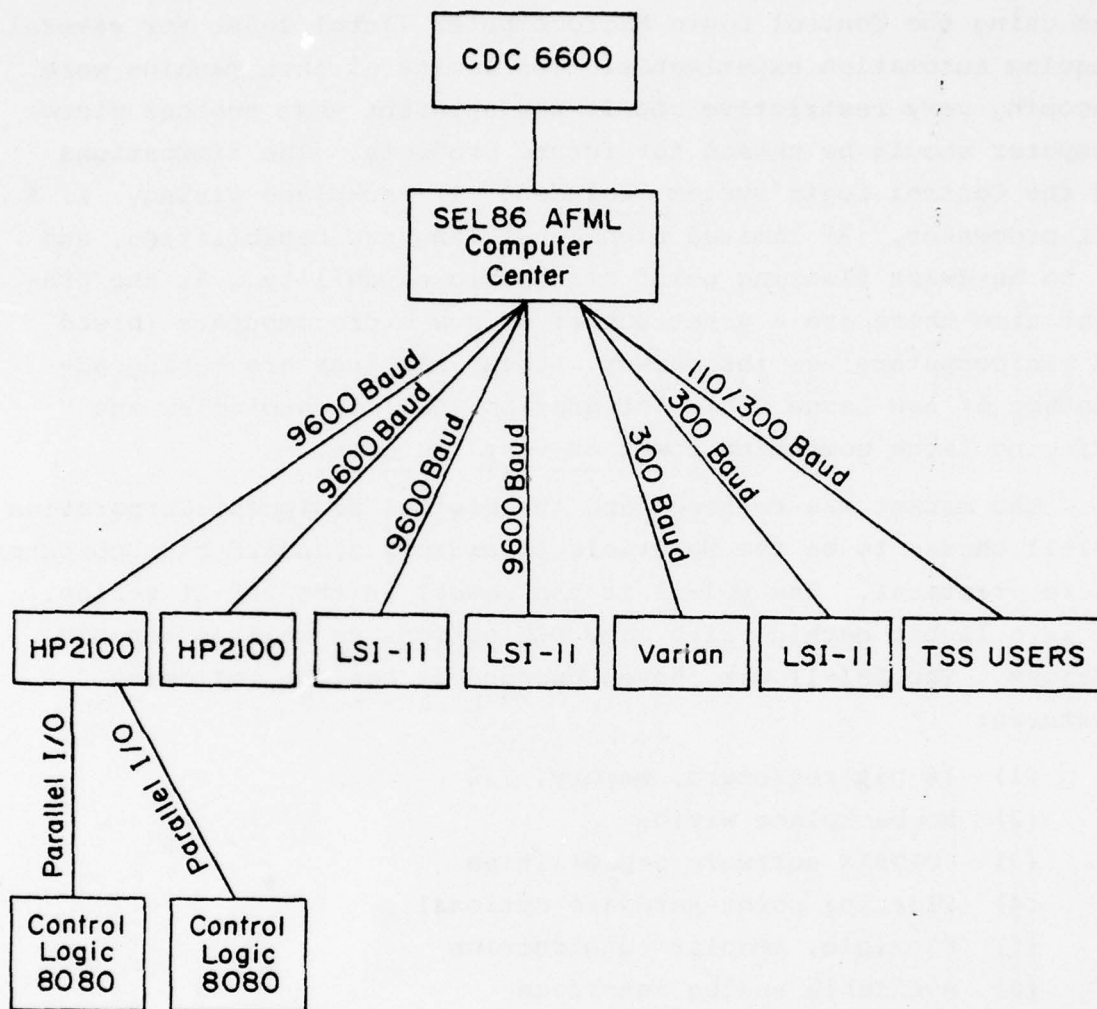


Figure 1. AFML Digital Computer Hierarchy.

2.2 SELECTION OF A LABORATORY STANDARD MICROCOMPUTER

At the beginning of this contract the Materials Laboratory was using the Control Logic Microcomputer (Intel 8080) for several ongoing automation experiments. The limits of this machine were becoming very restrictive and it was apparent that another microcomputer should be chosen for future projects. The limitations of the Control Logic system included: 1) Backplane wiring, 2) 8 bit processor, 3) limited high level language capabilities, and 4) no hardware floating point arithmetic capability. At the present time there are a great number of new microcomputers (based on minicomputers) on the market. These machines are taking advantage of new Large Scale Integration (LSI) technologies and offering large computing power at very low cost.

The market was reviewed and the Digital Equipment Corporation LSI-11 chosen to be the Materials Laboratory standard microcomputer where practical. The LSI-11 is the newest in the PDP-11 series. It is a 16 bit machine with many DEC options as well as non-DEC options. The LSI-11 was chosen because it has the following features:

- (1) 16 bit registers, memory, I/O
- (2) No backplane wiring
- (3) FORTRAN software capabilities
- (4) Floating point hardware optional
- (5) Flexible, modular construction
- (6) Available analog interface
- (7) DEC software support
- (8) DEC maintenance support
- (9) Reasonable cost (less than Control Logic)

At the time that the LSI-11 was chosen there were requirements for five data acquisition systems, and since that time additional proposals and preliminary designs using the LSI-11 have been generated. Due to the fact that it is a true state-of-the-art machine, the LSI-11 should remain as the laboratory standard microcomputer (where practical) for many years to come.

2.3 SPECIFIC APPLICATIONS

One of the activities of this contract has been the specification, design, and implementation of data acquisition and process control systems to automate the experiments in the Materials Laboratory. This activity has involved three computers: Control Logic (Intel 8080), DEC LSI-11, and HP 2100. The automation of experiments with these machines is intended to become part of the Materials Laboratory Computer Hierarchy (see Figure 1).

2.3.1 Liquid Adsorption Experiment Automation

The liquid adsorption experiment of the Surface Interactions Branch studies the rate of adsorption and desorption of a fibrous sample with various concentrations of solvent and solute liquids passing over it. Using a valve network as shown in Figure 2 pumps cause various concentrations of solvent and solute liquids to flow into the sample chamber and the adsorption of the sample is measured with the defractometer.

The liquid adsorption experiment of the Surface Analysis Group (AFML/MBM) has been implemented using a Control Logic Microcomputer. Figures 2 and 3 show block diagrams of this experiment and control system. A Technical Report (AFML-TR-76-XX) describes the experiment and automation.¹ This system (hardware and software) was initially fabricated by AFML/DOC personnel, but due to changes in assignments and manpower, the debug, redesign, and final implementation (hardware and software) were done under this contract. The software flowchart is shown in Figure 4. A listing of the program is given in Appendix A. The control system is used for two purposes - first to take the place of a human operator in overnight experiments, and secondly to automate the data gathering so that data can be input to a data reduction program in a more convenient form. The software for this experiment was generated using PLM, a high order language developed by INTEL Corporation for use on their 8080 microprocessor. AFML/DOC had access to the PLM compiler on the GE time-share system. The program was written, stored, and compiled on the GE time-share system, and the object file punched on paper tape to load into the Control Logic System.

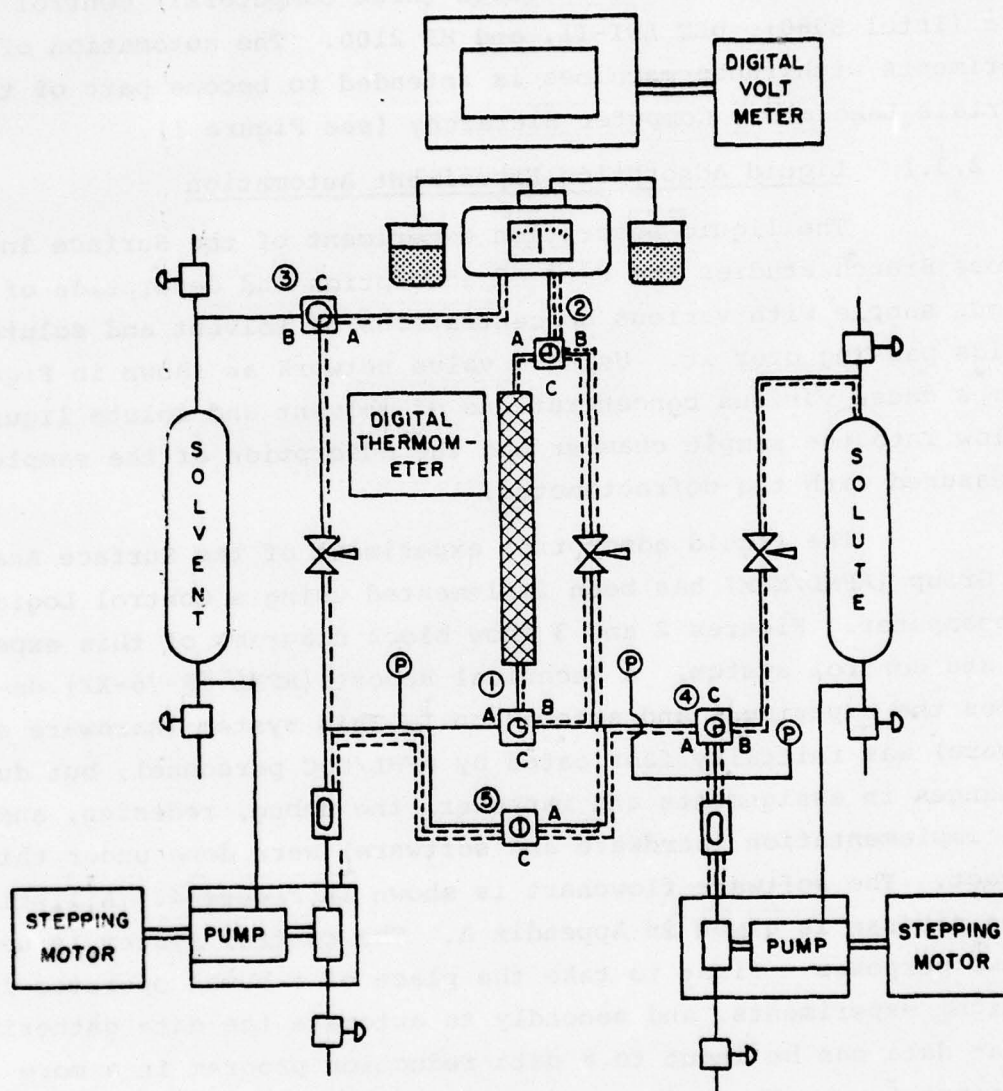


Figure 2. Surface Analysis Experiment Block Diagram.

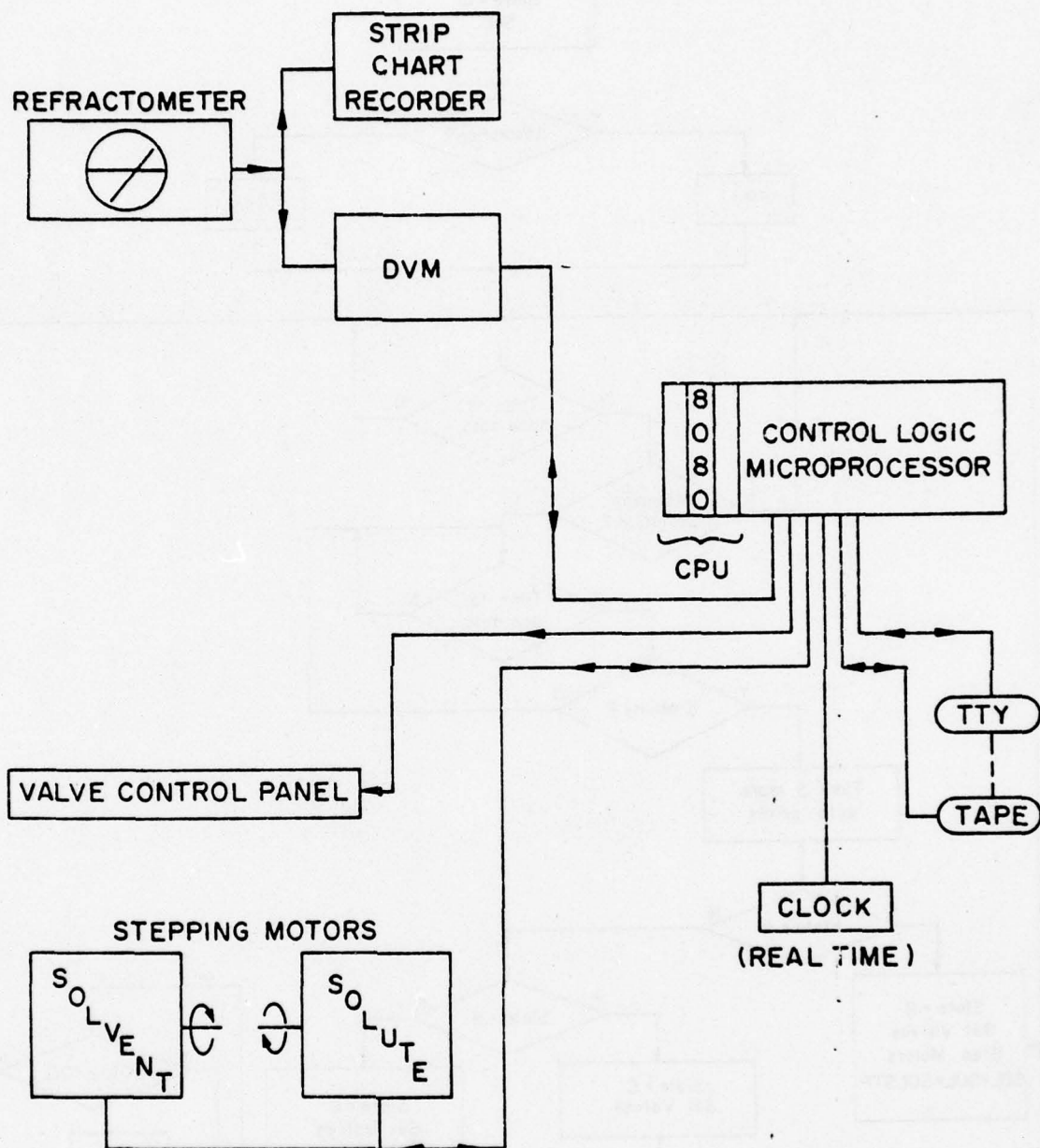


Figure 3. Control System for Surface Analysis Experiment.

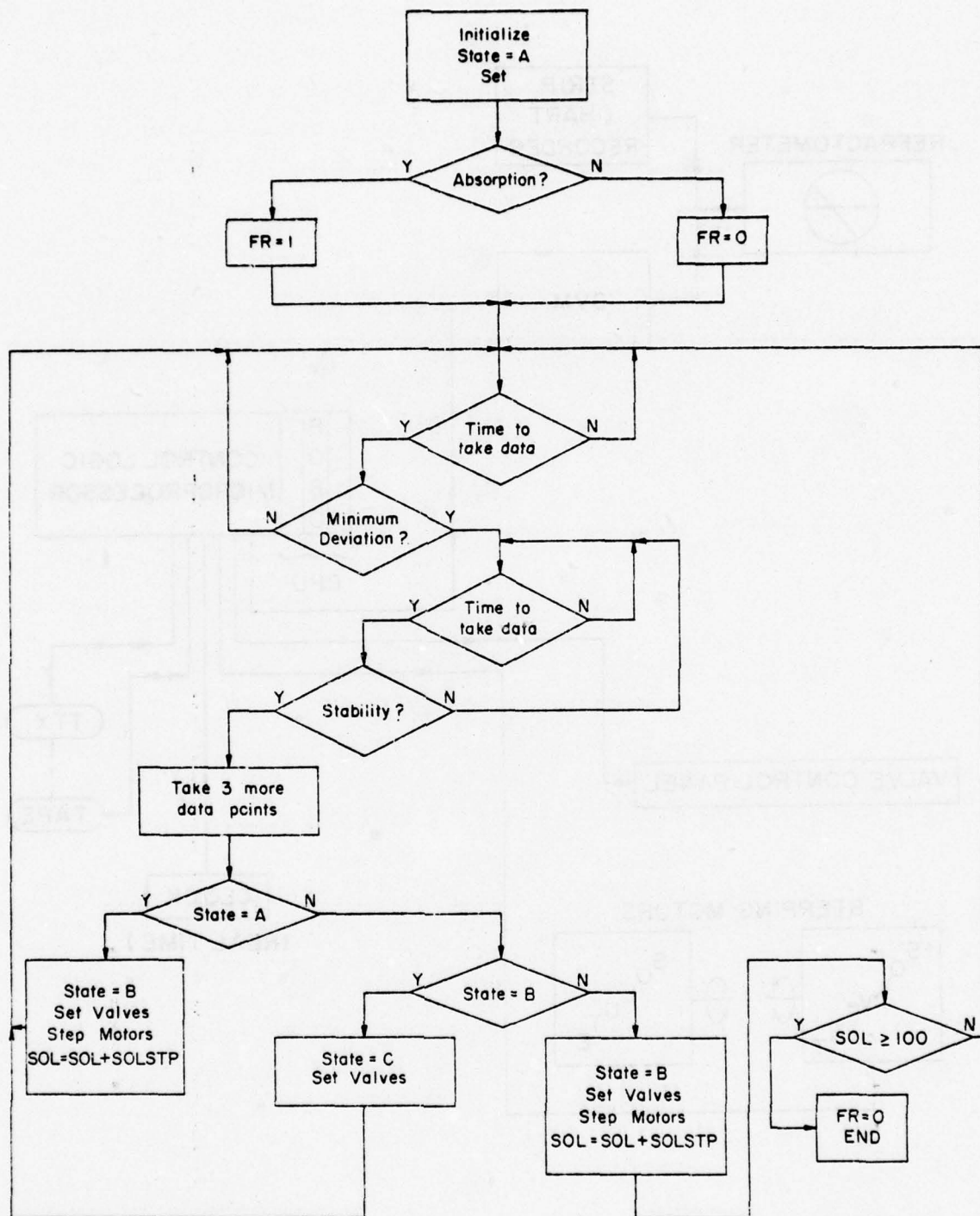


Figure 4. Software Flowchart.

The four states of the valves are called states A, B, C, and the rest state. These states relate to the quantity of solvent and solute passed over the sample. The rest state is the state in which the valves are not controlled by the computer, but rather by switches on the front panel. In state B a solution of solvent and solute is passed over the sample. In state C a solution of solvent only is passed over the sample. The solvent to solute ratio is stepped in increments from 0 to 1 and back to 0. The initial and final states of the experiment are state A. In state B and state C the interferometer measures the adsorption of the sample and the control system takes data from the interferometer, watching for the signal to stabilize. Once stability has been reached, the control system changes the state of the valves to the next state. Throughout this process the relative amounts of solvent and solute are changed by the control system through the stepping motors. In the past, one experiment required constant attention from an operator and the constant taking of data by an operator. Often one experiment could run for 48-72 hours. The computer control system has greatly relieved the operators task and made data gathering much easier.

2.3.2 Polymer-Surface Interaction Experiment Automation

The polymer-surface interaction experiment of the Surface Interactions Branch studies the gas volumetric adsorption properties of a material sample which is exposed to various concentrations of gases. The chamber containing the gases and sample is analyzed using a mass spectrometer. This automation required the control of a UTI-100C Mass Spectrometer by an HP 2100 Disk System minicomputer. The interface was configured using a Hewlett-Packard universal parallel interface and a Hewlett-Packard 8 channel Analog to Digital Converter. The block diagram of this automation system is shown in Figure 5. The parallel interface required some signal conditioning in order to drive some relays in the mass spectrometer. The cable and signal conditioner schematics are shown in Figure 6. The controls of the mass spectrometer such as operation mode, amplitude range, and sweep start are handled by the

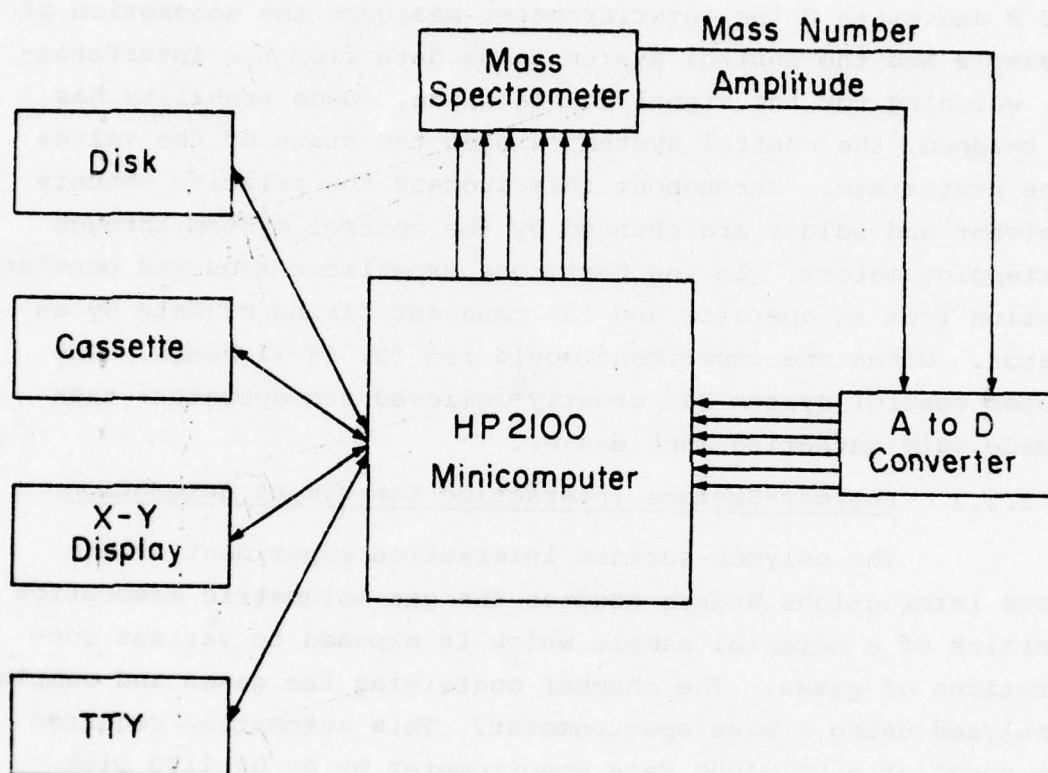


Figure 5. Automation of Polymer-Surface Interaction Experiment.

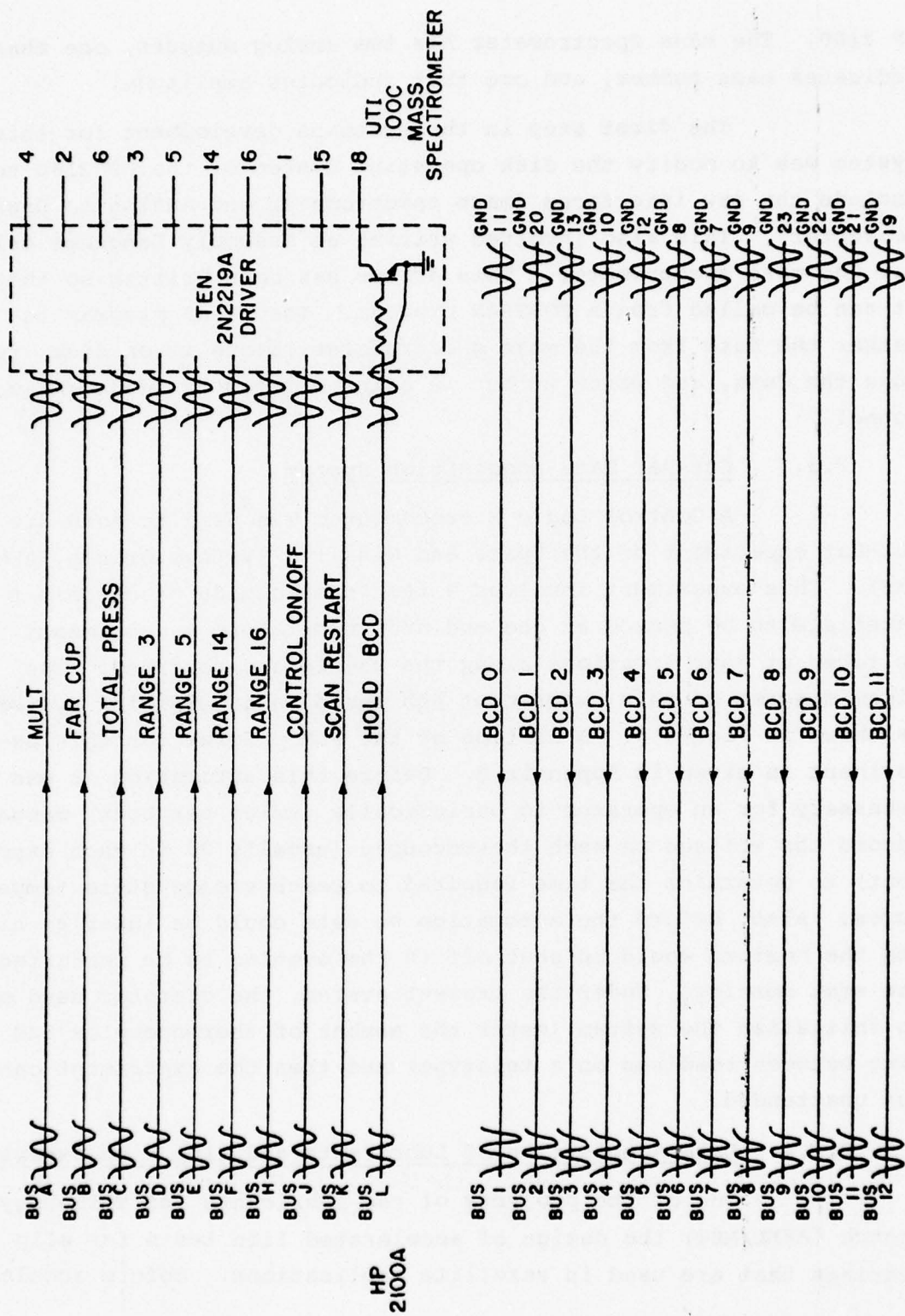


Figure 6. HP 2100 to Mass Spectrometer Interface.

HP 2100. The mass spectrometer has two analog outputs, one that indicates mass number, and one that indicates amplitude.

The first step in the software development for this system was to modify the disk operating system of the HP 2100 to include the new interfaces (mass spectrometer and Analog to Digital Converter). This also involved writing an Assembly Language driver for the mass spectrometer. This driver has been written so that it can be called from a FORTRAN program. The final program to gather the data from the mass spectrometer, store it on disk, reduce the data, and print it out is being written by AFML/DOC personnel.

2.3.3 Cut-Bar Data Acquisition System

A Control Logic Microcomputer was used to automate a Cut-Bar experiment of the Space and Missiles Systems Branch (AFML/MXS). This experiment involves a test sample made of different materials to be heated at one end and temperature measurements to be taken at many locations along the sample and recorded. The block diagram of this experiment and the data acquisition system is shown in Figure 7. A listing of the PLM program for this experiment is given in Appendix B. Before this automation it was necessary for an operator to periodically (twice per hour) manually record the voltage on each thermocouple (usually 20 in each experiment) to determine the time required to reach steady state temperatures. Also, before the automation no data could be taken at night and the heaters would be shut off in the evening to be restarted the next morning. Under the present system, the operator need only to initialize the system (enter the number of thermocouples and the time between readings on a teletype) and then the experiment can be run unattended.

2.3.4 Data Acquisition of Lubricants and Slip Ring Experiment

One of the projects of the Lubricants and Tribology Branch (AFML/MBT) the design of accelerated life tests for slip ring bearings that are used in satellite applications. Before accelerated

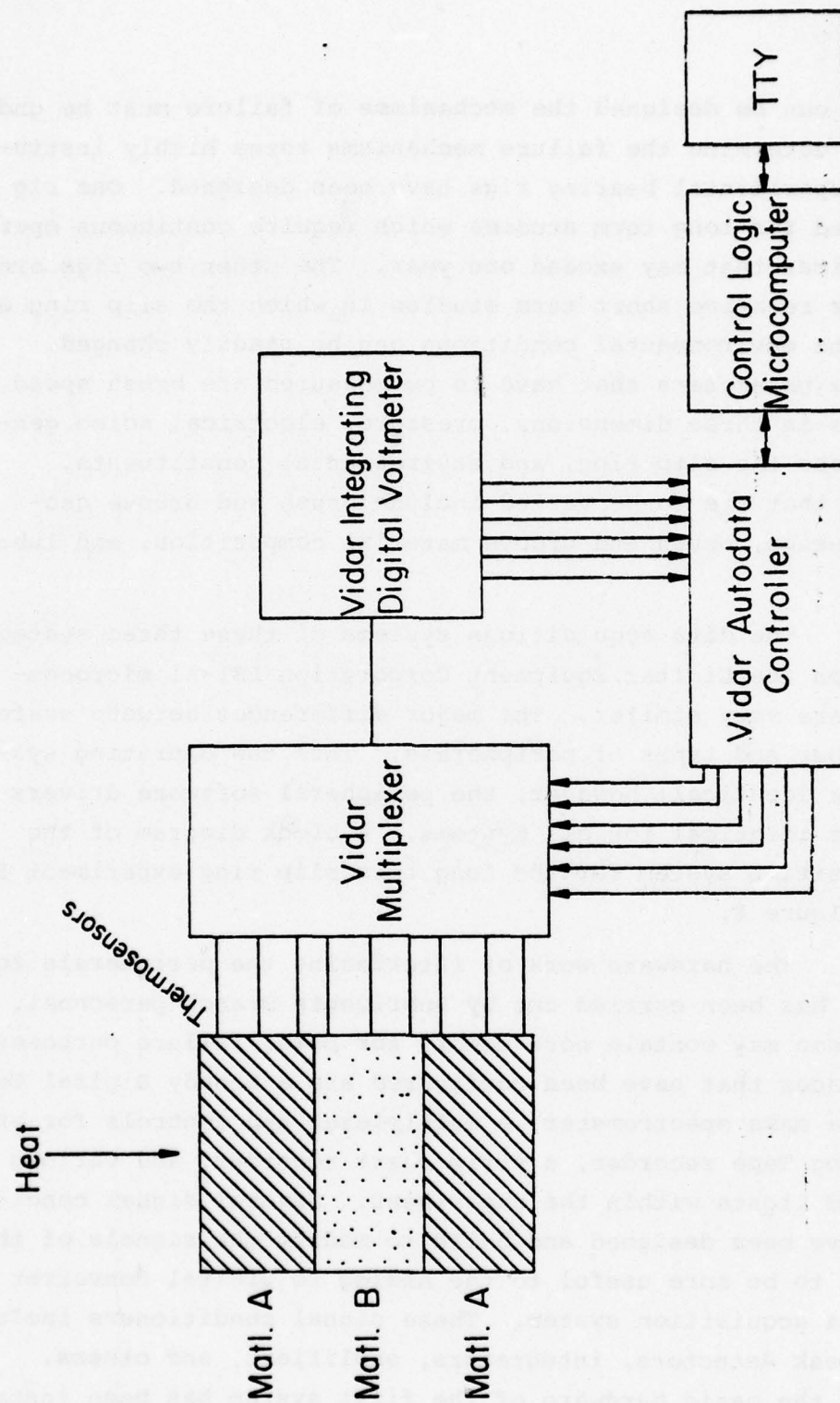


Figure 7. Block Diagram of the Cut-Bar Experiment.

life tests can be designed the mechanisms of failure must be understood. To determine the failure mechanisms three highly instrumented experimental bearing rigs have been designed. One rig will be used for long term studies which require continuous operation for times that may exceed one year. The other two rigs are to be used for relative short term studies in which the slip ring design, or the environmental conditions can be readily changed. Some of the parameters that have to be measured are brush speed, brush force in three dimensions, pressure, electrical noise generated across the slip ring, and environmental constituents. Parameters that are to be varied include brush and groove geometrical design, brush and groove material composition, and lubricants.

The data acquisitions systems of these three systems are based on the Digital Equipment Corporation LSI-11 microcomputer and are very similar. The major differences between systems is the number and types of peripherals. Thus the operating system will be identical; however, the peripheral software drivers will not be identical for all systems. A block diagram of the data acquisition system for the long term slip ring experiment is shown in Figure 8.

The hardware work of interfacing the peripherals to the LSI-11 has been carried out by Lubricants Branch personnel. These systems may contain core memory for power failure purposes. The interfaces that have been configured are a Kenedy Digital Tape recorder, a mass spectrometer, a multiplexer and controls for an Ampex Analog Tape recorder, a strip chart recorder, and various sensors and lights within the experiment. Several signal conditioners have been designed and build to modify the signals of the experiment to be more useful to the Analog to Digital Converter of the data acquisition system. These signal conditioners include filters, peak detectors, integrators, amplifiers, and others. Presently, the basic hardware of the first system has been installed, and the software drivers for the peripherals are being written to checkout the hardware. When these systems are complete it will

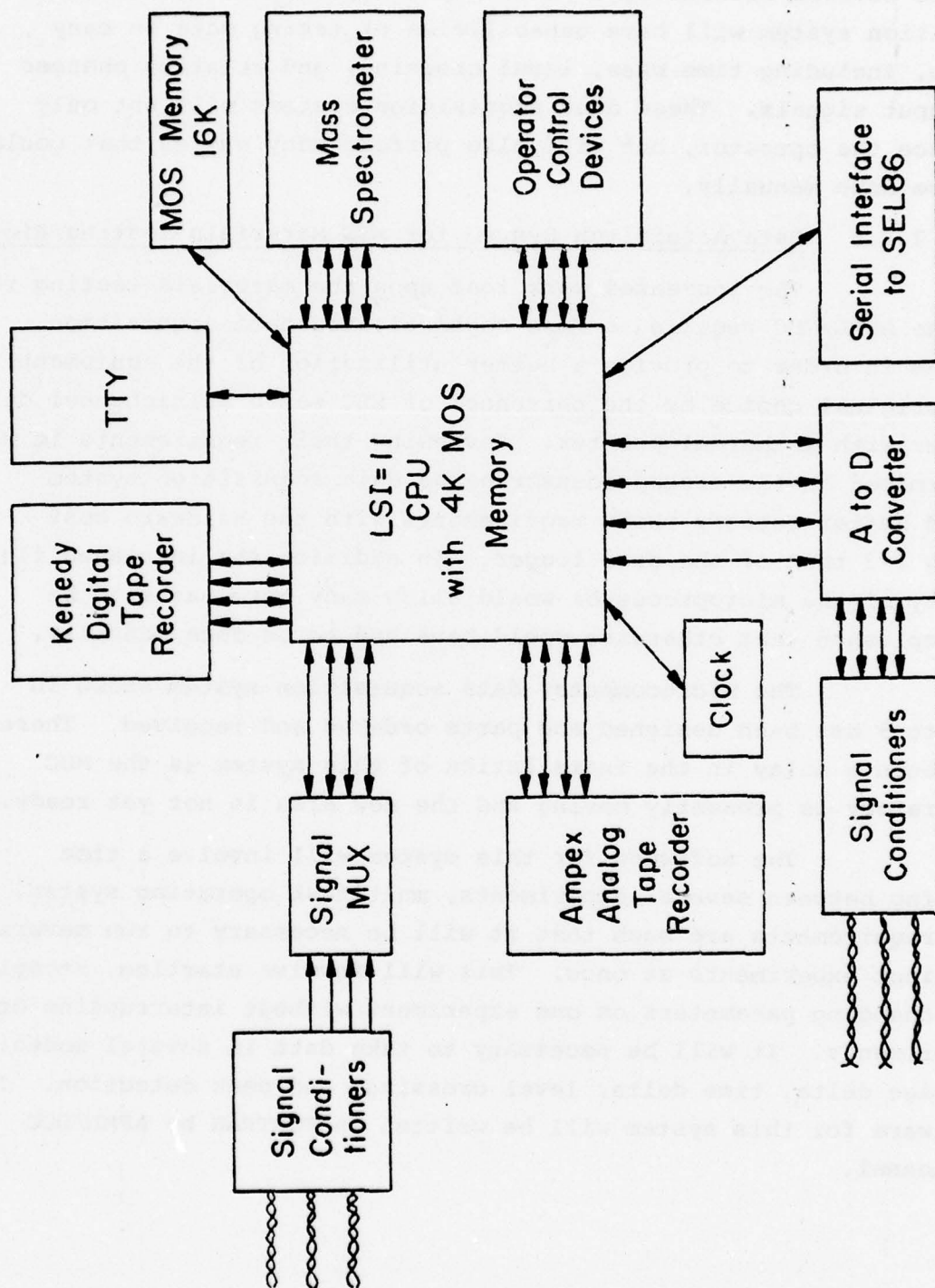


Figure 8. Lubricants Data Acquisition System Block Diagram.

be possible for the data acquisition systems to take data in any of the several methods on a 24 hour per day basis. The data acquisition system will have capabilities of taking data in many modes, including time base, level crossing, and relative changes of input signals. These data acquisition systems will not only replace the operator, but will also perform many duties that could not be done manually.

2.3.5 Data Acquisition System for MTS Materials Testing Rigs

The increased work load upon the materials testing rigs of the AFML/MBC required a more sophisticated data acquisition system in order to provide a better utilization of the equipment. The original choice by the personnel of MBC was a multichannel data logger with a thermal printer. Reviewing their requirements it was determined that a microprocessor based data acquisition system could better satisfy their requirements with the hardware cost being 2/3 that of the data logger. In addition the increased flexibility of the microprocessor would allow many more tasks to be accomplished that otherwise would have had to be done manually.

The microcomputer data acquisition system shown in Figure 9 has been designed and parts ordered and received. There has been a delay in the installation of this system as the MBC laboratory is presently moving and the new area is not yet ready.

The software for this system will involve a time sharing between several experiments, multitask operating system. The requirements are such that it will be necessary to run several distinct experiments at once. This will involve starting, stopping, and changing parameters on one experiment without interrupting other experiments. It will be necessary to take data in several modes: voltage delta, time delta, level crossing, and peak detection. The software for this system will be written in FORTRAN by AFML/DOC personnel.

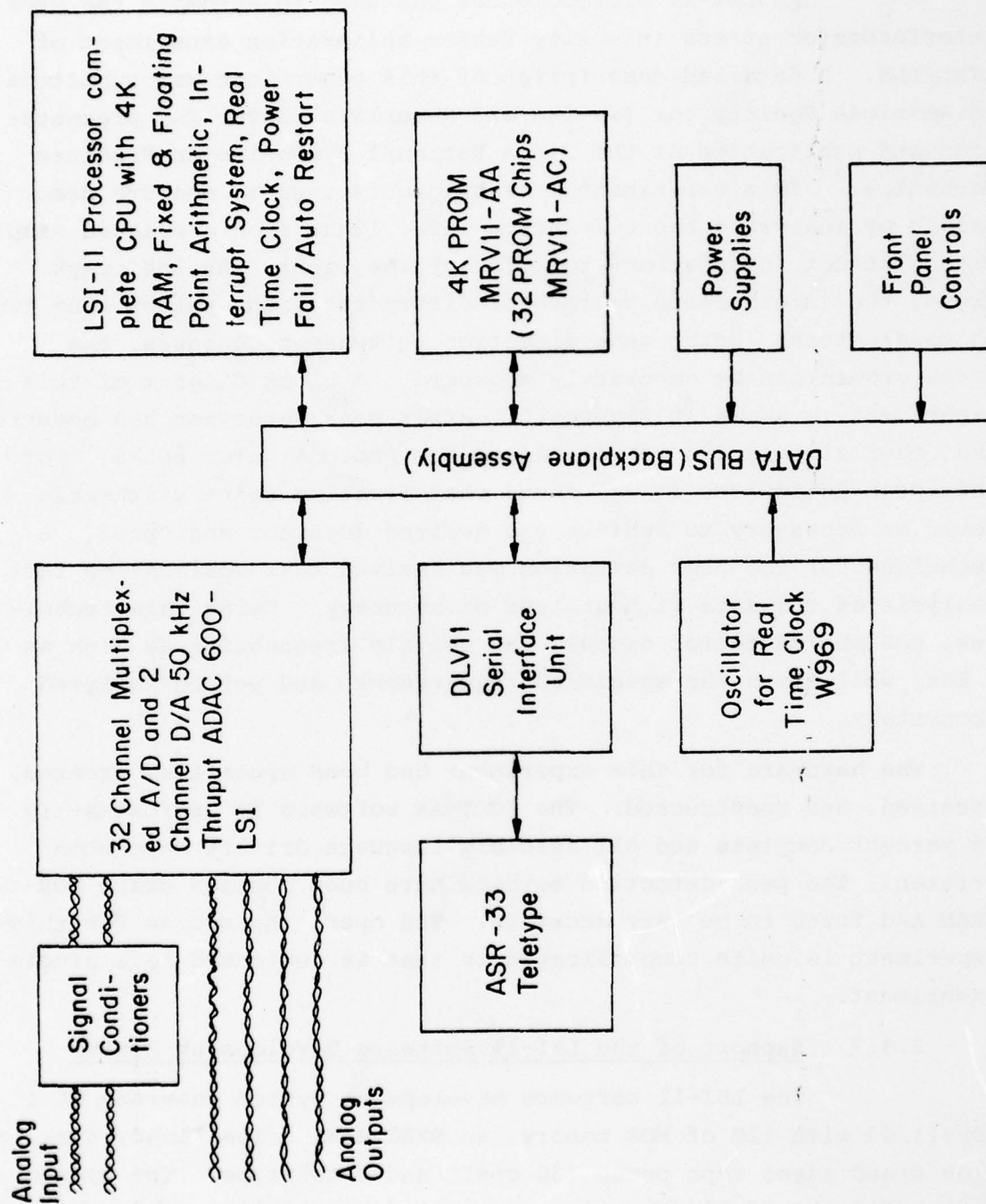


Figure 9. Data Acquisition System for the MTS Testing Rigs.

2.3.6 Automation of a Stress Intensity Factor Calibration Experiment

An LSI-11 microcomputer was used to automate the user interferometer stress intensity factor calibration experiment of AFML/LLN. A detailed description of this experiment was submitted to American Society for Testing and Materials (ASTM) for presentation and publication at the Ninth National Symposium on Fracture Mechanics.² This experimental technique is used to measure crack growth by analyzing the diffracted laser light from a cracked sample with distinct indentations to diffract the light. As the crack grows, the interference pattern of diffracted light passes over two photodetectors. Using peak detection software techniques, the crack growth can be accurately measured. A block diagram of this experiment is shown in Figure 10. After peak detection has occurred the other signals are correlated to the photodetector peaks. For this peak detection, it was found that floating point arithmetic would be necessary to achieve the desired accuracy and speed. A technique for the peak detection was derived that would allow fast analysis of the data without loss of accuracy. Using this technique, the photodetector signals can contain frequencies as high as 2 kHz, well above the specified requirement, and yet be analyzed accurately.

The hardware for this experiment has been specified, ordered, received, and constructed. The FORTRAN software is approximately 75 percent complete and all assembly language drivers have been written. The peak detection methods have been modeled using FORTRAN and found to be very accurate. The operating system for this experiment is quite uncomplicated in that is dedicated to a single experiment.

2.3.7 Support of the LSI-11 Software Development System

The LSI-11 software development system consists of a PDP-11/03 with 12K of MOS memory, an RX01 dual drive Floppy disk, a high speed paper tape punch (30 cps), and a teletype. The system runs under the RT-11 operating system and has FORTRAN IV capabilities.

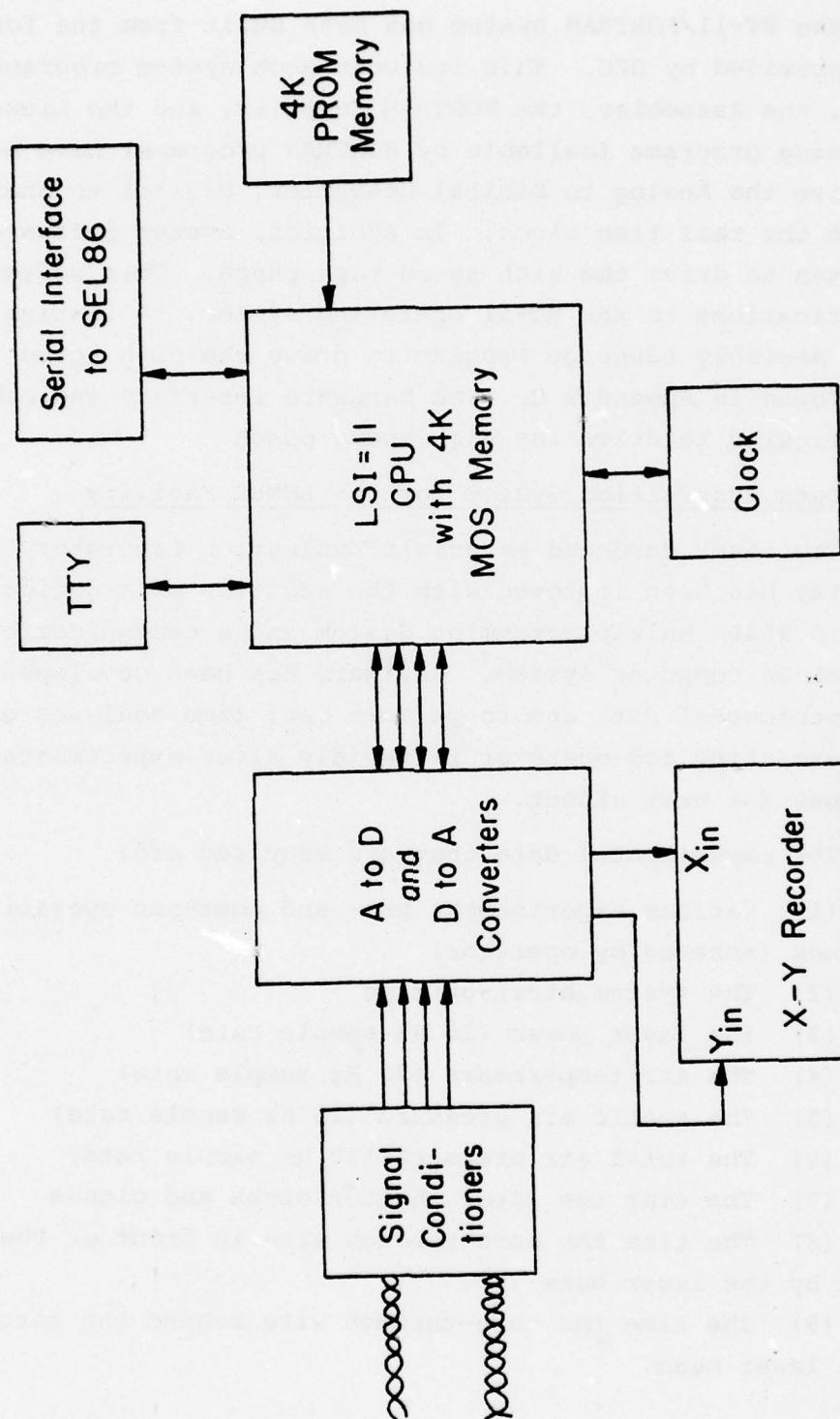


Figure 10. Block Diagram of Stress Intensity Factor Experiment Automation.

Our usage of the RT-11/FORTRAN system has been built from the four master disks provided by DEC. This includes such system programs as the Editor, the Assembler, the FORTRAN Compiler, and the Linker. Assembly Language programs (callable by FORTRAN programs) have been written to drive the Analog to Digital Converter, Digital to Analog converter, and the real time clock. In addition, system software has been written to drive the high speed tape punch. This software included modifications to the RT-11 operating system. A listing of the PDP-11 Assembly Language Program to drive the high speed punch can be found in Appendix C. The hardware interface and cables were also configured to drive the high speed punch.

2.3.8 Data Acquisition System for the LHMEF Facility

The Laser Hardened Materials Evaluation Laboratory (LHMEF) facility has been improved with the addition of a dedicated Hewlett-Packard 9640A Multiprogramming System and a communication link to the SEL 86 computer system. Software has been developed to acquire experimental data and to perform real time analyses of these data, permitting the operator to rapidly alter experimental parameter values for best effect.

The experimental data that are acquired are:

- (1) Various experimental pre- and post-run operating parameter values (entered by operator)
- (2) The system start-up time
- (3) The laser power (20 Hz sample rate)
- (4) The air temperature (20 Hz sample rate)
- (5) The static air pressure (20 Hz sample rate)
- (6) The total air pressure (20 Hz sample rate)
- (7) The time the laser shutter opens and closes
- (8) The time the burn-through wire in front of the target is cut by the laser beam
- (9) The time the burn-through wire behind the target is cut by the laser beam.

The principal component of the Hewlett-Packard system is a 21MX computer, operating under the Hewlett-Packard supplied RTE-C operating system. To begin an experimental run, the operator uses RTE-C to activate the program OZ, and enters the operating parameter values that have changed since the last experimental run. The operator then has the program OZGO activated to control the acquisition of the experimental data. Initialization of the laser control sequencer by the operator starts the program data acquisition sequence.

When the experiment terminates, the operator uses RTE-C to activate the program OXPST to enter the post-run operating parameters values. If desired the operator can have the program DADMP activated to inspect the raw data acquired during the run. If the data are acceptable, the SEL 86 computer program can be activated. Within this program the operator sends the raw data over a communication link to the SEL 86 computer and initiates the execution of the analysis program SELOZ (on the SEL 86). This program performs the following computations:

- (1) The run time (time of shutter closure minus system start-up time)
- (2) The shutter time (time of shutter closure minus time shutter opened)
- (3) The burn-through time (back burn-through wire time minus front burn-through wire time)
- (4) The laser power curve over the run time interval
- (5) The laser power threshold time (the time when the laser power becomes less than (or equal to) the laser power threshold going down minus the time when the laser power becomes greater than (or equal to) the laser power threshold going up)
- (6) The maximum, minimum, and average values of laser power over the shutter time interval
- (7) The maximum and average values of laser power over the laser power threshold time interval
- (8) The total energy over the laser power threshold time interval

- (9) The diameter and area of the beam on-target
- (10) The laser power on-target curve over the run time interval
- (11) The average power on-target over the shutter time interval
- (12) The average power density on-target over the shutter time interval
- (13) The energy density on-target over the shutter time interval
- (14) The average power on-target over the burn-through time interval
- (15) The average power density on-target over the burn-through time interval
- (16) The energy density on-target over the shutter time interval
- (17) The total temperature curve over the run time interval
- (18) The minimum temperature over the shutter time interval
- (19) The static air pressure curve over the run time interval
- (20) The RMS value of static air pressure over the shutter time interval
- (21) The total air pressure curve over the run time interval
- (22) The RMS value of total air pressure over the shutter time interval, and
- (23) The Mach number.

The data and analytical results are conveniently formatted and printed on a high speed printer. They are also archived on magnetic tape. CALCOMP quality plots are made of the power on-target, total temperature, static pressure, and total pressure curves. A summary of the analysis is printed for the operator on the LHMEI teletype. The operator can use this summary to select the experimental conditions for the next run. The users manual is found in Appendix D.

SECTION III
COMPUTER PROGRAMS FOR RE-ENTRY AND LASER APPLICATIONS OF
ABLATION/THERMAL EFFECTS

3.1 INTRODUCTION

During the last fifteen years many computational techniques have been developed to simulate the ablation/thermal effects occurring in nose tips, heat shields, and rocket nozzles. With some modifications these computer programs can also be applied to analyze the ablation/thermal response of materials to laser heating and provide guidelines in the development of new materials. AFML has sponsored two research programs to select appropriate computational techniques for ablation/thermal analyses. These programs have led to two sets of computer programs; Aerotherm Prediction Procedure for Laser Effects (APPLE), and Unified Ablative Material Thermal Response Analysis Procedure (UNIFIED). While these two sets of computer programs contain many similar mathematical algorithms, each has special features designed for their particular application. For instance APPLE permits in-depth absorption of radiant heating, while UNIFIED has additional codes for boundary layer calculations of re-entry environments and also a discrete model capability.

This section is divided into two parts; each part describes the application of one set of computer codes to current AFML problems. The application of the UNIFIED codes has been used to determine the sensitivity of a one dimensional thermal response solution of carbon phenolic in an arc-jet environment to perturbations of its thermophysical properties. Specific applications of the APPLE code include the heating of ZnSe and KCl CO₂ laser output coupler windows and simulation of a quartz lampbank heating of painted aluminum in a windtunnel.

3.2 APPLICATIONS OF THE APPLE CODE

3.2.1 Introduction

APPLE consists of a set of computer codes developed to describe material response to low power ($<10^5$ kw/cm²) laser radiation such as typified by an infrared CO₂ laser operating in the CW mode. The basis of these codes is the aerospace technology computer codes developed for combined radiative and convective heating conditions. The conversion of these codes to laser problems is feasible because low power heating rate is similar to that experienced by a high performance re-entry vehicle or a far planetary entry probe. Because these codes have been developed specifically for aerospace applications and not laser heating, a perfect transition is not expected in solving problems involving laser heating. Difficulties arising from the application of these codes are not due to numerical computational technique; but rather to the assumptions on thermophysical and thermochemical properties, necessitated by the lack of data on the material of interest.

The following descriptions are of the main routines comprising APPLE and are taken from the APPLE users manual.³

The Aerotherm Chemical Equilibrium (ACE) computer program is an extremely versatile code for calculating quantities of importance to a broad variety of thermochemical processes. The thermochemical processes treated may be divided into two categories; closed systems and open systems. Closed systems are those for which the relative amounts of each chemical elements in the system are constant if prespecified. Open systems are those for which the relative amounts of chemical elements depend on various mass transfer rates due, for example, to boundary layer convection or solid surface degradation. The ACE program can treat both systems in chemical equilibrium and systems for which certain reactions are kinetically controlled.* The most important objective of the

*The kinetic controlled reactions routine (KINET) was deleted for use in the APPLE computer program library.

the ACE program when used in conjunction with Aerotherm's CMA/CMAC and ASTHMA computer codes is to generate tables of gas-phase properties at the wall interface as a function of three independent variables: pressure, dimensionless gas blowing rate, and dimensionless char blowing rate (p , B_g' , and B_c' respectively). In addition to calculating the chemical and thermodynamic state of a variety of systems, the ACE program also calculates and prints out some transport properties (e.g., viscosity, thermal conductivity) appropriate to that state.

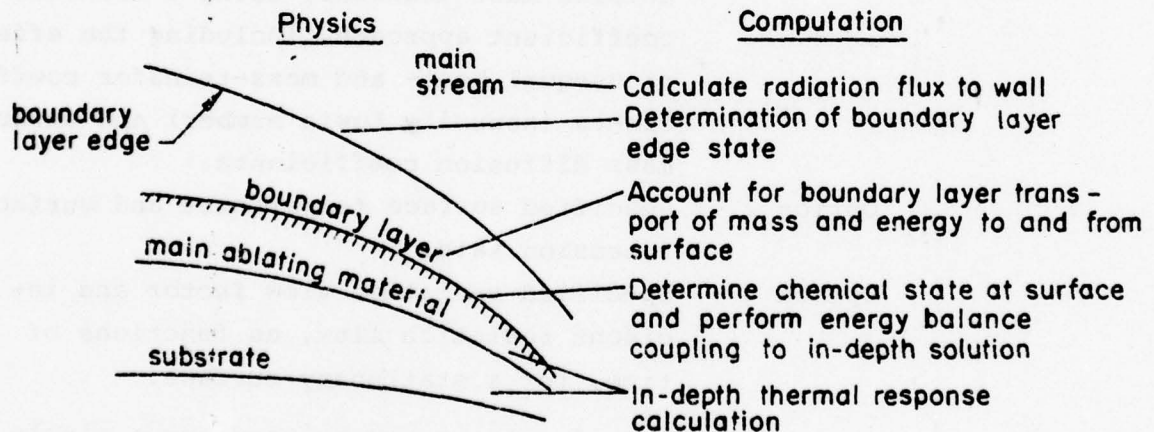
The Charring Material Thermal Response and Ablation Program (CMA) is an implicit, finite-difference computational procedure for computing the one-dimensional transient transport of thermal energy in a three-dimensional isotropic material which can ablate from a front surface and which can decompose in depth. Decomposition reactions are based on a three-component model. The program permits up to eight different back-up materials of arbitrary thickness. The back wall of the composite material may transfer energy by convection and by radiation.

The ablating-surface boundary condition may take one of three forms:

- Option 1 - General convection-radiation heating with coupled mass transfer, using a transfer coefficient approach, including the effects of unequal heat- and mass-transfer coefficients (nonunity Lewis number) and unequal mass diffusion coefficients.
- Option 2 - Specified surface temperature and surface recession rate.
- Option 3 - Specified radiation view factor and incident radiation flux, as functions of time, for a stationary surface.

Any combination of options may be used for a single computation. Option 3 is appropriate to cool down after termination of convective heat input and is often useful in conjunction with Options 1 and 2.

The Axi-Symmetric Transient Heating and Material Ablation Version 3 (ASTHMA3) Computer Program is a transient heat conduction program for two-dimensional, axi-symmetric bodies. Multiple non-charring, anisotropic materials may be studied. The surface boundary condition has three options, including an unusually general thermochemical erosion or ablation condition as well as simplified radiation and specified temperature options. The in-depth temperature prediction is of the familiar explicit finite-difference type. It allows a completely general finite difference mesh layout relative to the physical r - z axes, and accounts for anisotropic heat conduction effects. The heated surface boundary condition is an unusually general thermochemical type. It accounts for two specific kinetically controlled carbon reduction reaction, one kinetically controlled water gas shift reaction, and any number of gas-phase equilibrium reactions for any combination of ablating materials and environments. Chief applications for the computer program are rocket nozzles and entry vehicles. The following sketch serves to clarify various physical aspects of the ablation problem treated.



The JANAF routine is a special purpose code that has been incorporated into the COUPLE computer program. It assembles a molecular species thermochemical data (JANAF) file in the appropriate format for input into the ACE program. JANAF performs these functions by searching in the JANAF data tape for the molecular species that can be formed from a prescribed list of atomic elements that is specified in the input.

The COUPLE preprocessor routine is a user oriented program specifically developed to execute singly or coupled a variety of Aerotherm surface state thermochemistry and in-depth ablation and thermal response computer programs. It generates appropriate input data for the computer programs that are to be executed from simple execution directives and very little input data. Version 1 couples the ACE and JANAF routines to obtain surface state thermochemistry solutions that are used as an input to the CMA and ASTHMA computer programs. Version 2 couples the ACE, JANAF, and CMA routines, while version 3 couples ACE, JANAF, CMA, and ASTHMA routines.

While running APPLE coupled decreases total turn around time for analysis as only a single job is submitted, it is usually desirable to run APPLE uncoupled. Thus no parameter values are assigned by default and the validity of assumptions made can be more readily determined. In addition, if the problem doesn't run correctly, debug is easier as the responsible routines can more easily be located. Brief descriptions of problems solved with the APPLE code under this contract are included below. A more detailed description of these problems can be found in a series of AFML technical memos⁵⁻⁹, and in a technical report by Rondeau and Ford.¹⁰

3.2.2 Thermal Response of "Painted" Aluminum to Radiant Heating

Currently, various elastomeric coatings on aluminum plates mounted in a windtunnel are being exposed to radiant heating by a quartz lampbank. The objective of this experiment is the assessment of the nuclear flash protection afforded by the elastomeric coating applied to an aircraft's outer skin. A model of

of this experiment consisting of the thermophysical properties of a polyurethane coating material and aluminum 2024T3, the physical dimension, and the boundary conditions (heating rate and heat transfer coefficient) were used with the APPLE code to calculate the back face temperature of the aluminum substrate as a function of time. The agreement with experimentally obtained back face temperature was quite good.⁵ The use of APPLE does not supplant the need to do the experiment but rather compliments it. The experiment helps determine physical model to be used as input to APPLE, i.e., knowledge of whether the coating would peel, ablate, or blister during the experiment. Once agreement between experimental and calculated results is obtained, the APPLE code can then be used to predict the results for extrapolated experimental conditions.

In addition APPLE was used to parametrically study the effect of coating thermal conductivity on the surface and in-depth temperatures of a polyurethane coated aluminum sample subject to a thermal pulse in a windtunnel. The results show that temperature changes from a $\pm 20\%$ variation in the thermal conductivity of the coating are relatively minor.

The code was also used to model an in-flight scenario where the wing section of an aircraft, traveling at Mach 0.85 and 30,000 feet is exposed to the same heat flux as that produced in the laboratory experiment. Comparisons between the flight simulation and laboratory experiments have been presented along with the parametric study of the coating thermal conductivity in References 6 and 9.

3.2.3 Thermal and Stress Analysis of the AFWL Zinc Selenide Laser Window

Operating conditions of the Air Force Weapons Laboratory Electric Discharge Coaxial Laser (EDCL)¹¹ were used to calculate the axisymmetric (2D) temperature profiles in the zinc selenide output coupler CO₂ laser windows. These temperatures were then used to calculate the thermally induced stresses in the window. The calculated stresses were well within design expectations for the given operating conditions.⁷

3.2.4 Computed Temperature Response of the AFML Flat-Top Laser Window

AFML's 10 Kilowatt Flat-Top laser is based on the design of AFWL EDCL laser.¹¹ Because of the greater absorptance of the AFML zinc selenide windows, as determined by post-mortem examination, sufficiently high thermally induced stresses resulted in two window failures under normal operating conditions. The APPLE codes were used to verify that thermally induced stresses were a possible failure mechanism and to generate curves of maximum thermally induced stress versus time for various power levels. These calculations provide safe operating criteria to prevent failure of the laser coupler window. This work is described in detail in Reference 8.

3.3 UNIFIED COMPUTER PROGRAMS

3.3.1 Introduction

The primary objective in Unified set of computer programs as developed by Aerotherm is to apply the advanced computer technology developed over the past 15 years (for the ablation design analysis problem) to the far more difficult problem area of materials development guidance. The relative payoffs of altering the chemical composition of a composite material, reductions of density and thermal conductivity, and changing the composites configuration both on a macroscale and a microscale are all important areas that can be addressed with the Unified set of codes. The one dimensional computer program, CMA, primary use has been to determine the sensitivity of material response in an arc jet to changes in the thermophysical properties of carbon phenolic. This data coupled with the relative uncertainties of the individual, thermophysical properties illustrate where the most beneficial use of scarce resources can be used to reduce the uncertainty of the parameters that most effect the material response. Similar analyses can be done for a re-entry environment and using the two dimensional computer program ASTHMA4; however the more interesting

results will be using ASTHMA4 to obtain material responses due to material configuration. The first study using CMA is described below.

3.3.2 Sensitivity Study Using CMA

A Carbon Phenolic button, backed by a stainless steel plate, placed in an arc jet was modeled for analysis with the CMA computer code. Fourteen nodes were placed in the Carbon Phenolic button and two nodes in the stainless steel backup plate. The environmental conditions used were those similar to a typical arc-jet experimental test run (i.e., pressure = 1.07 atm, heat input of 1050 BTU/(ft²-sec) for 30 seconds and 0 for the remaining 70 seconds). The baseline thermophysical properties of carbon phenolic that were used are shown in Table I. The properties that were varied $\pm 10\%$ each are shown in Table II.

The response, as a function of time, of the carbon phenolic material was measured by seven parameters. These are the interface temperatures between the carbon phenolic and the stainless steel and 6 dimensionless variables shown in Table III.

Thirty-five computer runs were made (2 runs as each parameter was varied $\pm 10\%$ and the baseline case). Since the material response functions as defined in Table II are not calculated directly by the CMA computer program, the CMA computer program was modified to create a new file which contained all quantities necessary to calculate these response functions. This file was catalogued and then at a later time used as an input file to the computer program RES which then calculated the individual response function. This computer program is listed in Appendix E.

The sensitivity coefficients, defined as $\frac{\Delta R_j}{\Delta X_i}$ where ΔR_j is the change in the j^{th} material response function due to a ΔX_i change made to the X_i material input parameter, are determined by the computer codes. The sensitivity coefficients for R_0 (the interface temperature) and the experimental uncertainty in the materials input parameters are listed in Table IV.¹² The

TABLE I
THERMOPHYSICAL PROPERTIES OF CARBON PHENOLIC
(CCAZ CLOTH/5C 1008 RESIN)

MATERIAL	TEMPERATURE (°R)	SPECIFIC HEAT (BTU/LB-°R)	CONDUCTIVITY (BTU/LB-FT-°R)	EMISSIVITY
Virgin Carbon Phenolic (MX4926, 0° Layup)	530	0.25	1.45×10^{-4}	.85
	800	0.34	1.65	.85
	1160	0.38	1.97	.85
	1500	0.38	2.34	.85
	6000	0.38	2.34	.85
Charred Carbon Phenolic (MX4926, 0° Layup)	530	0.16	0.64×10^{-4}	.85
	1000	0.33	0.64	.85
	1500	0.39	0.64	.85
	2000	0.42	0.64	.85
	2500	0.45	0.78	.85
	3000	0.47	0.95	.85
	3500	0.49	1.1	.85
	4000	0.51	1.3	.85
	5000	0.53	1.4	.85
	6000	0.55	1.5	.85
Stainless Steel (Backface Material)	492	.11	22.2×10^{-4}	.14
	672	.11	26.1	.14
	1032	.11	30.3	.14
	1400	.11	30.4	.14

PYROLYSIS GAS ENTHALPY		INTERNAL DECOMPOSITION KINETIC DATA				
TEMP °R	H BTU/LB	i	ρ_{v_i} lb/ft ³	ρ_{c_i} lb/ft ³	B_i sec ⁻¹	E_i °R
900	-1899.0					
1800	-761.9	1 (resin)	19.34	0.	1.4×10^3	1.54×10^4
2700	8868.6					
3600	2894.7	2 (resin)	58.00	29.24	4.8×10^9	3.68×10^4
4500	3953.5					
5400	5747.2	3 (reinforcement)	98.55	98.55	0.	0.

HEAT OF FORMATION OF PHENOLIC: -363 BTU/LB

VIRGIN MATERIAL RESIN MASS FRACTION: 0.330

VOLUME FRACTION: 0.378

TABLE II
THERMOPHYSICAL PROPERTIES THAT WERE VARIED $\pm 10\%$ EACH SEPARATELY

VIRGIN DENSITY OF FABRIC	CLEAR SPECIFIC HEAT OF COMPOSITE AS f(TEMP)
GLASS DENSITY OF FABRIC	EMISSIVITY OF VIRGIN COMPOSITE AS f(TEMP)
VIRGIN DENSITY OF RESIN 1	EMISSIVITY OF CHARRED COMPOSITE AS f(TEMP)
VIRGIN DENSITY OF RESIN 2	BLOWING RATE PARAMETER
GLASS DENSITY OF RESIN 2	PYROLYSIS GAS ENTHALPY AS f(TEMP)
VOLUME FRACTION OF RESIN IN VIRGIN COMPOSITE	PYROLYSIS REACTION GAS COUNTS FOR RESIN 1
VIRGIN CONDUCTIVITY OF COMPOSITE AS f(TEMP)	PYROLYSIS REACTION GAS COUNTS FOR RESIN 2
CHAR CONDUCTIVITY OF COMPOSITE AS f(TEMP)	HEAT OF FORMATION
VIRGIN SPECIFIC HEAT OF COMPOSITE AS f(TEMP)	

TABLE III
MATERIAL RESPONSE PARAMETERS (R_j)

$$R_0 = T_{BF}$$

$$R_1 = \frac{\int_0^X \frac{k}{\rho C_p X^2} dx}{\int_0^X dx}$$

$$R_4 = \frac{\int_0^t h_g \dot{m}_g dt}{\int_0^t \dot{q}_{in} dt}$$

$$R_2 = \int_0^t R_1 dt$$

$$R_5 = \frac{\int_0^t h_x \dot{\rho} dt}{\int_0^t \dot{q}_{in} dt}$$

$$R_3 = \frac{\int_0^t \dot{q}_{out} dt}{\int_0^t \dot{q}_{in} dt}$$

$$R_6 = \frac{\int_0^t H^T \dot{m}_g dt}{\int_0^t \dot{q}_{in} dt}$$

T_{BF} - INTERFACE TEMPERATURE

k - THERMAL CONDUCTIVITY

ρ - DENSITY OF THE MATERIAL

C_p - SPECIFIC HEAT OF THE MATERIAL

X - THICKNESS OF THE MATERIAL

\dot{q}_{in} - RATE OF HEAT ABSORBED BY THE MATERIAL

\dot{q}_{out} - RATE OF HEAT LOST BY THE MATERIAL

h_g - ENTHALPY OF THE GASEOUS MATERIAL

\dot{m}_g - RATE OF MASS LOSS OF THE MATERIAL

h_x - ENTHALPY OF THE SOLID MATERIAL

H^T - HEAT OF FORMATION

t - TIME.

TABLE IV
SENSITIVITY COEFFICIENTS FOR INTERFACE TEMPERATURE

I	PARAMETER	COEFFICIENT			PRODUCT
		$\frac{\Delta Ro/\Delta X_i}{\ln X_i}$	UNCERTAINTY		
1.	ρ	1.0480	.02		.021
2.	ρ_{cf}	-1.2590	.02		.025
3.	ρ_{vf}	0.0284	.01		---
4.	ρ_{v1}	-0.0515	.01		---
5.	ρ_{v2}	0.0083	.02		---
6.	ρ_{c1}	0.0256	.05		.001
7.	τ_f	0.2626	.10		.026
8.	$k_v(T)$	0.1570	.20-?		.031
9.	$k_c(T)$	-0.1080	.10		.010
10.	$C_{pv}(T)$	0.0322	.15-?		.005
11.	$C_{pc}(T)$	-0.0009	.02		.002
12.	$E_v(T)$	-0.0695	.02		.001
13.	$E_c(T)$	-0.0110	.10-?		.001
14.	λ	0.0109	.05		.001
15.	$H_g(T)$	0.1275	.05		.006
16.	E_i, B_i	0.2270	.05		.01
17.	E_i, B_i	0.1287	.02		.002

total uncertainty of a material response function ΔR_j is given by

$$\Delta R_j = \sum_i \frac{\Delta R_j}{\Delta X_i} \Delta X_i \quad (1)$$

with $\frac{\Delta R_j}{\Delta X_i}$ being the sensitivity coefficient and ΔX_i the experimental uncertainty of the i^{th} material input parameter. An appreciable reduction in ΔR_j can be made by reducing the uncertainty ΔX_i only if the product involving ΔX_i in Equation (1) is large in comparison to other terms in the sum. The parameters that most significantly effect the uncertainty of the interface temperature as determined by this study and listed in Table IV are:

- | | | |
|-----|-----------------------|---------------------|
| (1) | CHAR CONDUCTIVITY | - MOD SENS/HIGH UNC |
| (2) | VIRGIN CONDUCTIVITY | - MOD SENS/HIGH UNC |
| (3) | CHAR FABRIC DENSITY | - HIGH SENS/MOD UNC |
| (4) | VIRGIN FABRIC DENSITY | - HIGH SENS/MOD UNC |
| (5) | VIRGIN SPECIFIC HEAT | - MOD SENS/MOD UNC |
| (6) | REACTION CONSTANTS | - MOD SENS/MOD UNC |

Similar analyses are being conducted for the other response characteristics and are to be published.

SECTION IV
COMPUTER ANALYSIS OF MICROSTRUCTURES

4.1 COMPUTER RECOGNITION OF TITANIUM ALLOW MICROSTRUCTURES

4.1.1 Introduction

The objective of this project is to predict the mechanical properties of titanium alloys by computer examination of photomicrographs of their microstructures. Prior to this contract, computer codes were developed to effect the solution.¹³ The results, however, were inconclusive; the correlations obtained did not consistently provide accurate predictions of the mechanical properties of samples processed. The previously developed solution was basically three steps.

1. A digital image of a microstructure is produced by digitizing a photomicrograph negative. The digital image consists of light intensity readings (pixels) taken at 40 micron intervals of the negative. The pixel (picture points) intensities are valued 0-255.

2. The digital image is operated on by computer software, the particles of the microstructure are extracted and their geometric characteristics determined.

3. These geometric characteristics of each image are correlated with the measured mechanical properties of the respective Ti-alloy samples, yielding a relation that will predict mechanical properties from the geometric characteristics of the particles in the microstructure of an alloy sample. Work performed on this project under this contract was concerned primarily with improvements in the computer processing involved in Part 2 above.

4.1.2 Initial Work

The first step taken in working on the project was to become familiar with the problem and gain a working knowledge of the use of the computer codes used to process the digital images.

Soon after work was underway on the project, a new set of photomicrographs of the Ti-alloy surfaces, which were of higher quality than photomicrographs used in the previous work, became available. It was theorized that processing these would yield better correlation results, and the processing of the new images was begun. Several of the programs that performed initial image processing were found to be inoperative. It was determined that the majority of the problems encountered in the attempt to execute these programs were due to changes in the operating system of the CDC 6600 computer that had been made since the programs were written and last used in a production mode. The computer codes were modified to make them compatible with the present CDC 6600 operating system and the processing of the new images continued.

4.1.3 Improvements of the Digital Processing

A close examination of the available intermediate results of the processing done during previous work on the project was undertaken in an effort to gain insight into possible improvements that could be made. Conclusions reached were that image thresholding and particle separation/extraction should be the areas of refinement. The digital image thresholding, in which the raw image pixel values of 0-255 are reduced to a value of 0 or 1, is the most critical stage of the processing; improvements in this process would cause significant improvement in the accuracy of the particle geometric characteristics determined at the final stage of the image processing. In the particle separation/extraction phase, it was observed that frequently particles that were very close together in the photomicrograph were extracted from the digital image and treated as one large particle. This was due to higher density levels of the pixels in the area of the junction of these particles, caused by light scattering and the resultant noise when the photomicrograph was digitized.

The two operations of image thresholding and particle separation, however, are not totally independent of each other. It was seen that the accuracy of particle separation/extraction would be greatly facilitated by a more accurate thresholding of the digital images. The approach taken to the improvement of the image processing was the development of a more accurate thresholding process, and the later development of a more precise method of particle separation/extraction, if necessary.

4.1.4 Improvement of Image Thresholding

The initial step in establishing an improved thresholding process was the collection of data of raw image characteristics. This data consisted primarily of computer printouts of the digital images and corresponding plots of pixel density versus frequency for a selection of photomicrographs.

In the previous method, the threshold density D_T was determined by

$$D_T = (D_H - D_L)/2$$

where D_L was the density below which five percent of the pixels of the image were contained, and D_H the density above which five percent of the pixels were contained. (This represents the standard technique for thresholding used in image analysis.) Through comparison of the original photomicrograph, the printout of the raw image, the density versus frequency plot, and the printout of the thresholded image for each sample used, it was concluded that this method did not consistently yield the best thresholded image.

Several alternate thresholding methods were examined. The techniques evaluated first were differential, line-by-line thresholding methods, and failed to produce accurately thresholded images. The poor results obtained are believed to be due to overall density variation across the photomicrograph, relatively high-density variation within the particles themselves, and general inaccuracies associated with applying one-dimensional operations to two-dimensional structures.

The thresholding method arrived at was essentially a modification to the original method. The thresholding density D_T was determined as

$$D_T = (D_H - D_L) * D_I$$

where D_H and D_L remained unchanged from the original method, and D_I is the percentage of the remaining image pixels to be considered as being within a particle. D_I is an input parameter to the thresholding program. Using a carefully selected value of D_I , determined by examining the density histogram for each image processed, the respective thresholded images were improved significantly.

4.1.5 Improvements in Particle Separation/Extraction

During the processing of the new photomicrographs, the particle separation/extraction portion of the computer program was found to be totally inoperative. Due to the program's complexity and lack of documentation, it was necessary to examine the entire code in detail in an effort to locate and correct the error(s). In doing so, it was discovered that numerous major programming errors were present; that in fact when operative the codes would not perform the operations as described in the furnished documentation. Because of these conditions, the design and development of new particle separation/extraction software was required. Also, the magnitude of the errors was certain to have adversely affected the results previously obtained using these codes.

4.1.6 Development of New Particle Separation/Extraction Software

Several methods of particle separation/extraction were investigated, in order to select the most accurate one for implementation. The first methods investigated employed circular and rectangular "windows" of various sizes which were "moved" over the raw digital images; while examining the portions of the image contained in the window. Particle edges are detected using this method by looking for large density variations occurring within

the window. Once such a density change is found, the window is then moved slowly around the particle by maintaining the density differential interior to the window while shifting its position on the image, thus outlining a particle. This approach failed, however, due to the wide range of particle sizes, making the selection of the proper window area by the computer program extremely difficult at best. (Often a window would entirely enclose a small particle in one part of an image, and then be entirely enclosed itself by a large particle in a nearby portion of the image.) Separation of two seemingly connected particles was unpredictable using this method, due to the varying sizes of these "bridges" connecting the particles in the digital image.

The method of particle separation/extraction chosen to be implemented employs the same basic concepts as the original theory.¹³ In addition, the software incorporates the following:

1. input parameters should provide the capability of selecting the exact separation criteria for the case of erroneously connected particles;
2. actual separation should not be biased with respect to particle location in the image (previous method was biased in an upper-left sense, due to processing the image left to right and top to bottom);
3. the software should be much more efficient in terms of central memory required than its predecessor;
4. intermediate states of the image during separation/extraction should be available for precise evaluation of software performance.

The separation/extraction operation was broken into two main steps, to be performed by two individual programs:

1. Generation of interior particle pixel values (1, 2, or 3, depending upon the location of a pixel with respect to the outer boundary of the particle).
2. Separation and extraction of the particles.

The design, development, and testing of a program (PARSEP) to perform the first step has been completed. This program requires much less central memory for its execution (55,000₈ words versus 165,000₈ words for the previous program). This improvement was made possible by maintaining a much smaller portion of the image in central memory during processing. In addition, a utility program (IM6PRN) was written to print the image during and after the processing for software evaluation purposes.

The programming of the second step to process the image output by PARSEP (separation and extraction) was divided into five parts:

1. separation and extraction of particle cores (3-valued pixels);
2. determination, separation (when necessary), and extraction of the upper particle extremes (2-valued pixels);
3. determination, separation (when necessary), and extraction of the lower particle extremes (2-valued pixels);
4. determination, separation (when necessary), and extraction of the remaining left and right particle extremes (2-valued pixels); and
5. reconstruction and output of the particles.

The determination and separation operations stated above are effected by moving a window of dimensions 1 pixel by 1 pixel around in the stated area of a particle. This method locates particles from their interiors radially outward. In the case of seemingly "connected" particles, the user may specify the exact location at which these particles will be separated. This separation actually takes place at the user-specified relative location between the particle cores (3-valued pixels). For example, the user may specify that "connected" particles be separated at the location halfway between their cores. Software to perform parts 1, 2, and 3 has been designed, developed, and tested. Software to perform steps 4 and 5 has been designed and is presently in the development stage.

4.1.7 Additional Software Developed for Testing and Evaluation Purposes

In the process of testing and evaluating both previously existing and newly developed software, the following programs were written:

1. PICTURE - A program to print a "picture" of the raw digital images using 32 grey levels, obtained by overprinting of characters, for the purpose of quick, easy comparison of the raw digital image to the photomicrograph used to create it.
2. PARPLOT - A program to reconstruct and graphically display a microstructure from the geometric characteristics output by the MEAD program PROPERT.
3. SAMPLOT - A program to produce plots of frequency versus the log of area for particles extracted from images in previous work on the project.
4. FREQPLT - A program to produce plots of the image density histograms produced by the MEAD program HISTCDC.

In addition to the work described above, other work performed in association with this project consisted of the following:

1. Preparation of 10 image tapes used by the PAR Corporation in work related to this project.¹⁴
2. Familiarization with the use of the OLPARS (On-Line Pattern Analysis and Recognition System) of the Rome Air Development Center at Griffiss Air Force Base for the purpose of determining geometric characteristics-mechanical properties correlations. This system was determined suitable to perform correlation analysis when the new image geometric data becomes available.

4.2 COMPUTER MODELING OF CRACK PROPAGATION IN THE SURFACE TI-ALLOYS

4.2.1 Problem Description

In work performed with AFML/LLS, an effort to develop computer codes to model crack propagation across the surface of Titanium alloys was initiated as part of a continuing effort in this area. Given an alloy surface, the project objective is to, use the computer to predict the nature of a crack growing across it. At the present, this alloy surface is presented as an enlarged photograph of an alloy surface in which the features affecting crack propagation (grain structure, large particles, etc.) are easily discernable to the naked eye. Crack propagation is to be effected according to propagation-surface conditions relations specified by the software user.

4.2.2 System Requirements

Initial analysis of the problem yielded the following requirements for the developed software for the modeling process.

1. Capability of accurate representation of the alloy surface.
2. Capability of easy modification to that surface representation.
3. A high degree of flexibility in allowing for the specification of crack behavior criteria with respect to surface conditions, yet perform in strict accordance to that criteria.

Due to the desirability of a high degree of user-software interaction, the software developed functions in an interactive mode. There existed the need for graphics capability, for both establishing and modifying the surface representation, and for displaying intermediate and final results of the modeling process. AFML's TEKTRONIX 4014 graphics terminal and 4954 graphics tablet in conjunction with the INTERCOM system of the CDC 6600 was selected to be used since this configuration satisfies the interactive graphics capabilities required.

4.2.3 Computer Representation of the Alloy Surface

The photograph of an alloy surface to be processed is used to establish the digital surface representation in the computer. This consists, basically, of five steps:

1. Prior to execution of the modeling program, the boundaries of areas of homogeneous grain orientation on the photograph are manually outlined. Within each outlined area, a line determining the grain orientation to be used for that area is drawn.
2. Each boundary and respective grain orientation line on the photograph are digitized using the TEKTRONIX 4954 graphics tablet.
3. The output from the digitizer is stored in the computer. The computer surface representation then, in essence, consists of a set of lines.
4. The surface representation is displayed graphically on the terminal screen. Any bad points resulting for erroneous digitized input are corrected by editing.
5. The final step is to eliminate gaps between points on the boundary lines, to insure that a crack cannot pass between two points on a boundary line and not be recognized by the computer as having intersected that boundary line. This is done by converting boundary line point values to integers and interpolating points between the original points comprising a boundary line such that for two consecutive points on a boundary line (X_1, Y_1) , (X_2, Y_2) ,

$$|X_2 - X_1| = 1 \text{ or } 0,$$

and

$$|Y_2 - Y_1| = 1 \text{ or } 0.$$

In any case, one of the above relations must have a value of 1.

The above operations, with the exception of 1, are performed by programs of the CRAKMOD software system.

4.2.4 Linkage Table Generation

Once the user is satisfied with the established surface representation, the table of point-to-point links used to determine crack propagation is generated by CRAKMOD. At the present stage of development, these links are determined solely from the grain orientations of each boundary. (The linkage table structure, however, is designed to provide for additional links to be determined by additional crack propagation-surface condition criteria, as these relations become known.)

4.2.5 Crack Propagation

The propagation by the computer of a crack across the represented surface is initialized by the CRAKMOD user specifying its starting location. The software then determines the path of the crack by chaining together the points specified by the linkage table.

The main output from CRAKMOD is the graphical display of the surface and the crack grown across it. (Optional display capabilities include magnified views of one or more user selected surface areas, with or without the segments of the crack (if any) across them displayed.)

At the present stage of development, the CRAKMOD user may effect different crack configurations by specifying changes in the starting location of the crack and the grain orientations for the areas enclosed by the boundary lines. As other surface condition/crack propagation relations are recognized, all that will be necessary to modify CRAKMOD for their implementation will be the addition of program modules to generate the proper links, and provide for the input of any additional necessary features from the photograph (via digitizer or terminal). CRAKMOD is completely modular in structure, such that the implementation of the above features will require a minimum amount of modification to the existing software.

4.2.6 Additional Capabilities

In addition to the generation of cracks across the surface representation, CRAKMOD is capable of processing a crack digitized from the alloy surface photograph. Thus, crack generated by CRAKMOD may be compared to an actual crack in the surface processed.

There is the capability of measuring the length of the crack very accurately; much more accurately than is possible by measuring directly from a photograph or an actual sample, because the measurement may be made along the length of the actual crack using CRAKMOD, rather than along a straight line from its initial point to its end.

A listing of all FORTRAN programs developed for computer analysis of microstructures is enclosed in Appendix F.

SECTION V

GENERAL USER PROGRAMS AND EDUCATION

5.1 INTRODUCTION

A number of general user programs were written for use by AFML personnel. These include programs designed for graphics, digitizing, data analysis, support of the LSI-11 microprocessors and support of the Control Logic microprocessors. In addition, classes for AFML CDC 6600 and LSI-11 users were held to improve the utilization of these computers.

5.1.1 General User Programs for the CDC 6600

DIGIPLT is an easy-to-use, completely interactive software package that provides the capability of digitizing, displaying in graphic and tabular form, editing, and interpolating, data curves. Digitization and Graphics require a Tektronix 4394 or 4395 graphics tablet and 4014 Terminal respectively. The other tasks can be done at any terminal.

A complete description and instruction on DIGIPLT is given in the DIGIPLT USERS GUIDE in the Appendix G of this report.

In work performed under this contract, two relatively small, specific-purpose data processing programs were written. These programs are listed here along with a brief description of each.

DATPLOT - A program to produce log-log plots, with axis systems, of crack growth rate versus stress intensity.

CONVRT - A program to convert data output from the electron microprobe to usable format; edit erroneous points; and compute the mean and standard deviation for each data set processed.

These programs are listed in Appendix H.

5.1.2 LSI-11 Peripheral Driver Programs

The DEC RT-11 operating system can be used to generate stand alone FORTRAN program which contain a driver for a terminal. To drive other peripherals (such as A/D converter, D/A converter, real time clock, and modem interface) it is necessary to write Assembly Language programs that can be called from FORTRAN programs. There is a linking scheme used by RT-11 FORTRAN for Assembly Language subroutines. This scheme was used in writing these programs. The input and output parameters of the subroutines and the FORTRAN calls are shown in Table V. Listings of these PDP-11 Assembly Language programs can be found in Appendix I.

5.1.3 PLM Support Programs

The PLM compiler (Intel 8080 high-level language) originally used by AFML/DOC was the compiler installed on the General Electric Time-Share System. This compiler had two steps of operation. The first step had input of an ASCII source PLM program file and output of an ASCII file that contained Assembler Language and compiler error information. The second step had as input the output file from the first step and output of an ASCII file that contained memory and symbol information as well as the memory loading information. It was necessary to punch an ASCII paper tape of the memory loading information and convert that information to a binary paper tape suitable for loading by the paper tape loader supplied by Control Logic. This conversion had been done previously on a Control Logic System with low-speed tape reader and was very time consuming (45 minutes for a 4K program). In addition, there was noise on the telephone lines which made punching a correct ASCII tape very unlikely. Many times the tape would have to be punched four or five times to get an error free tape. A FORTRAN program was written for the GE system that used the ASCII memory loading file as input and created a binary output file compatible with the paper tape loader. A listing of this program is in Appendix J. The binary output file was then dumped to the terminal tape punch (30 cps). This process greatly improved

TABLE V
CALLING ASSEMBLY LANGUAGE SUBROUTINES
FROM RT-11 FORTRAN

Analog to Digital Converter:

CALL ADC (ICH, IN)

where ICH = Channel number to be converted (0 to 31)
IN = Integer equivalent of analog input on
A/D channel number ICH

Digital to Analog Converter:

CALL DAC (ICH, IOUT)

where ICH = Digital to Analog converter number (1 or 2)
IOUT = Integer equivalent of desired voltage
on D/A channel number ICH

Clock:

CALL CLOCK (IFLG, NHR, NMIN, NSEC, NMSEC)

where IFLG = 0 for read clock,
1 for set clock
NHR = Integer number of hours
NMIN = Integer number of minutes
NSEC = Integer number of seconds
NMSEC = Integer number of milliseconds

turnaround time for minor program changes, especially in the debug procedure. An option was added to print an octal symbol table rather than the standard hexadecimal symbol table.

Also a FORTRAN program was written to scan through the output file of the first compiler step and print errors. A listing of this program is in Appendix J. Previously it has been necessary to search this file for errors line by line, which was not only slow and difficult, but inaccurate. This program would print each line with an error, the error diagnostic, and the total number of errors. Both of these utility programs will be adopted to either the SEL 86 or the CDC 6600 for future PLM use.

5.2 AFML USER EDUCATION

5.2.1 CDC 6600 Users

To determine methods AFML could employ to realize more efficient utilization of allocated computer resources, an analysis of AFML's use of the CDC 6600 was made. The results of this analysis indicated the following would yield a significant increase in efficiency and throughout for prime shift processing:

1. Limiting the Computer Resource Units available per AFML prime shift job to 100 CRUS. This would prevent a small number of large jobs expending a large percentage of AFML's allocated resources, which then results in low throughout long turnaround AFML jobs for the remainder of the shift.
2. AFML computer users requesting only the resources required for their computer jobs versus gross overestimates, resulting in unnecessary restrictions on computer resources available to AFML while such jobs are in the input queue or in execution.
3. More efficient use of permanent file space by using all five available cycles per permanent file; and by maintaining on permanent file only information currently being processed.

To effect the above, it was seen that a user education effort was in order to explain and demonstrate to AFML users the

ways such practices could be carried out and how their use would benefit the individual user. A class was conducted during which methods of efficient use of the computer were demonstrated and discussed.

A second analysis of AFML computer use was made to evaluate the effectiveness of the effort. The results showed an increase of 50% in prime shift throughout, and a significant increase in available permanent file space.

5.2.2 LSI-11 Users

A two day class attended by 14 people was held on the applications of the LSI-11 microcomputer for potential LSI-11 users. A wide variety of subjects were discussed ranging from very general to very specific. A large emphasis was placed on I/O as this applies to all users.

APPENDIX A
LIQUID ADSORPTION EXPERIMENT PLM PROGRAM


```

2870 GO TO STEPL;
2880 SS30K: = SHL(SS(0),4);
2890 SS(0) = SS(0) - 2600;
2900 SS(1) = SS(1) - 2600;
2910 SS(0) = SS(0) + SS(1);
2920 /* TIME STEP SETUP */
2930 TIMES: CALL CRLF;
2940 DO P = 0 TO LAST (M);
2950 CALL OUTM(1);
2960 OUTPUT (M) = OT (P);
2970 END;
2980 DO P = 0 TO 1;
2990 A(P) = TS(P) + 2600;
3000 CALL OUTM(1);
3010 OUTPUT (8) = A(P);
3020 END;
3030 CALL CRLF;
3040 DO P = 0 TO 1;
3050 CALL INM(1);
3060 A(P) = INPUT (0);
3070 CALL OUTM(1);
3080 OUTPUT (8) = A(P);
3090 IF A(P) = SLASH THEN GO TO SETSDUM;
3100 END;
3110 DO P = 0 TO 1;
3120 TS(P) = A(P) - 2600;
3130 END;
3140 CALL TIMES;
3150 SETSDUM:
3160 DO P = 0 TO 4;
3170 CALL CRLF;
3180 END;
3190 END;
3200 DO P = 0 TO 2;
3210 DEF SL(P) = 0;
3220 END;
3230 JFLG=1;
3240 CALL DVM$IO;
3250 JFLG=0;
3260 O1=(DEF(0)/16)*1000;
3270 O2=(DEF(0) AND 170)*100;
3280 O3=(DEF(1)/16)*10;
3290 O4=(DEF(1) AND 170);
3300 LAST=DI+O2+O3+O4;
3310 SSAB=(LAST+4)/8;
3320 HTS=IS(0)*10+IS(1);
3330 H1(2)=720;
3340 H1(5)=720;
3350 H1(8)=400;
3360 H1(9)=400;
3370 RETURN;
3380 END SETUP;
3390 /* DIGITAL VOLT METER SUBROUTINE.....*/
3400 /* DATA IS GATHERED WITH THE MOST SIGNIFICANT BCD DIGIT
3410 PLACED IN DVM(0), AND THE LEAST SIGNIFICANT DIGIT IN DVM(2).

```

```

100 DECLARE (T,M) BYTE,
110 DEF(S) BYTE, /* PRESENT DEFRACOMETER READING */
120 DEFSL(S) BYTE, /* LAST DEFRACOMETER READING */
130 SOL(S) BYTE, /* SOLUTE LEVEL */
140 SS(S) BYTE, /* SOLUTE STEP */
150 TS(S) BYTE, /* TIME BETWEEN SAMPLES 00-99 MINUTES */
160 HT(S) BYTE,
170 SLASH LITERALLY '2570',
180 STRL BYTE, /* 1=STARTLED DUA READING, 0=ERRPR */
190 AA LITERALLY '0010',
200 AB LITERALLY '3020',
210 CC LITERALLY '3030',
220 /* NEW VALVE CONTROL TECHNIQUE */
230 /* PORT 32 BITS 1 AND 0(LSB'S)=VALVE #1
240 PORT 32 BITS 5 AND 2 =VALVE #2
250 PORT 32 BITS 5 AND 4 =VALVE #3
260 PORT 32 BITS 7 AND 6(MSB'S)=VALVE #4
270 PORT 33 BITS 1 AND 0(LSB'S)=VALVE #5
280 VALVE STATE A=10
290 VALVE STATE H=01
300 VALVE STATE C=00
310 VALVE NUMBER 1 2 3 4 5
320 STATE
330 INITIAL C C R A B C
340 A H R A B A
350 H C C C A A
360 C H A C A A
370 INITI2 LITERALLY '01010000',
380 INITI3 LITERALLY '00000000',
390 AA32 LITERALLY '011001010',
400 AA33 LITERALLY '0000001010',
410 AB32 LITERALLY '100001001',
420 AB33 LITERALLY '0000001010',
430 CC32 LITERALLY '100010010',
440 CC33 LITERALLY '0000001010',
450 /*IS BYTE, /* 1=FORWARD (ADSORPTION), 0=DESORPTION. */
460 FSR BYTE, /* 1=FORWARD (ADSORPTION), 0=DESORPTION. */
470 DUA BYTE,
480 (D1,D2,D3,D4,D5,D6,D7,D8,D9) ADDRESS,
490 IFLG BYTE,
500 (A4,C14) BYTE,
510 (PV,LV) ADDRESS,
520 DELTA BYTE,
530 (A1,A2,SSAUS,STRCHK) BYTE,
540 STAFF BYTE, /* KEEPS PRESENT STATE OF VALVES. */
550 FLAG BYTE,
560 JFLG BYTE,
570 PUS LITERALLY '2550',
580 NEG LITERALLY '2550',
590 E BYTE, /* E=1 IF ENTERING DESH., E=2 IF PROG. FINISHED. */
600 POLM BYTE,
610 /* TTY I/O WAITING SUBROUTINES .....*/
620 /* IT IS NECESSARY TO CHECK THIS BIT SO THE PROCESSOR
630 WILL KNOW WHEN A CHARACTER HAS BEEN PUNCHED ON THE TTY.

```

03:44:11 16:00:00 PICTURE2

```

640 THE SECOND HIT IS CHECKED TO SEE IF THE TTY IS READY TO
650 ACCEPT THE NEXT CHARACTER.
660 INWAIT: PROCEDURE?
670 DO WHILE NOT ROL (INPUT (1,3))
680 ENDO
690 RETURN?
700 ENDO
710 /AUG?
720 OUTWAIT: PROCEDURE?
730 DO WHILE NOT ROL (INPUT (1,1))
740 ENDO
750 RETURN?
760 ENDO
770 /* CP AND LF SUBROUTINE .....*/
780 /* THIS OUTPUTS THE PROPER CHARACTER TO THE TTY FOR A
790 CHARACTER RETURN AND A LINE FEED.
800 EXIS: PROCEDURE?
810 CALL OUTWAIT;
820 OUTPUT (3) = 150; /* CR */
830 CALL OUTWAIT;
840 OUTPUT (3) = 212; /* LF */
850 RETURN?
860 ENDO
870 /* NO MS WAITING SUBROUTINE .....*/
880 /* THIS IS A GENERAL UTILITY SUBROUTINE. FOR THIS
890 PROGRAM IT IS USED TO ALLOW THE HP DVM DATA TO SETTLE
900 AFTER ENCODING, BEFORE READING THE NEW VALUE.
910 PROCEDURE?
920 DECLARE II BYTE;
930 DO II = 1 TO 10;
940 CALL TIME (250);
950 ENDO
960 RETURN?
970 END JRSSEC;
980 /* SETUP SUBROUTINE.....*/
990 /* SETUP IS THE FIRST ROUTINE CALLED UP UNDER PROGRAM
1000 CONTROL. IT ASKS THE OPERATOR FOR THE VALUES NAMED:
1010 C= (SOLUTE LEVEL THE PROGRAM IS TO START AT)
1020 S= (SOLUTE STEP, OR WHAT PERCENTAGE OF 100 WE WISH
1030 THE STEPPING MOTORS TO JUMP EACH TIME.)
1040 L= (NUMBER OF MINUTES TO WAIT BEFORE TAKING NEXT DATA POINT.)
1050 A SLASH MARK MEANS (VALUE IS OK, CONTINUE)
1060 DVMSTIME: PROCEDURE?
1070 DECLARE (J,K) BYTE, PER LITERALLY '2500';
1080 A(3) BYTE,
1090 n(3) BYTE,
1100 SUM(3) BYTE,
1110 (SUM1, SUM2, SUM3) BYTE,
1120 (A1, A2, A3) BYTE;
1130 /* GATHERING 10 VALUES FROM DVM FOR AVERAGE. */
1140 DEFBL(0)=DEF(0);
1150 DEFBL(1)=DEF(1);
1160 DEFBL(2)=DEF(2);
1170 SI: DO J = 0 TO 9;
1180 OUTPUT (16) = J;
1190 CALL JRSSEC;

```



```

1200 CALL TIME(250);
1210 CALL TIME(250);
1220 CALL TIME(250);
1230 DO K = 0 TO 2;
1240   DO CASE K;
1250     A(K) = 0;
1260     A(K) = INPUT (14);
1270     A(K) = INPUT (19);
1280   END;
1290   A(K) = SHL(A(K),4);
1300   DO CASE K;
1310     H(K) = INPUT (17);
1320     H(K) = INPUT (18);
1330     H(K) = INPUT (19);
1340   END;
1350   U(K) = SHR(H(K),4);
1360   A(K) = A(K) + H(K);
1370 END;
1380 IF J < 0 THEN GO TO ADD;
1390 DO K = 0 TO 2;
1400   SUM(K) = A(K);
1410 END;
1420 GO TO NDJ;
1430 /* TAKING THE AVERAGE DVM READING */
1440 ADD: A1 = A(0);
1450 A2 = A(1);
1460 A3 = A(2);
1470 SUM1 = SUM(0);
1480 SUM2 = SUM(1);
1490 SUM3 = SUM(2);
1500 SU4(P) = DEC(A3 + SUM3); /* DECIMAL ADDITION */
1510 SUM(1) = DEC(A2 PLUS SUM2); L1520 SUM(0) = DEC(A1 PLUS SUM1);
1520 NDJ:
1530 END;
1540 /* NOW PLACE THE AVERAGE DVM READING INTO DEF. */
1550 DO J = 0 TO 2;
1560   DEF(J) = SUM(J);
1570 END;
1580 LV=PV;
1590 D1=(DEF(0)/16)*1000;
1600 D2=(DEF(1)/16)*100;
1610 D3=(DEF(2)/16)*10;
1620 D4=DEF(1) AND 170;
1630 PV=D1+D2+D3+D4;
1640 /* OUTPUT SUM ONTO TTY. */
1650 IF JFLG=1 THEN GO TO DONE;
1660 NO J = 0 TO 2;
1670   U(J) = SUM(J); /* VERY NECESSARY STEPS AFTER EXIT FROM */
1680   SUM(J) = H(J); /* THE DEC COMMAND. */
1690   H(J) = SHR(U(J),4);
1700   H(J) = H(J) + 2600;
1710 CALL OUTWAIT;
1720 OUTPUT (8) = H(J);
1730 IF J = 0 THEN
1740   DO;
1750

```

```

1760 CALL OUTWAIT;
1770 OUTPUT (A) = PER;
1780 END;
1790 SUM(J) = SUM(SOL(J),4);
1800 SUM(J) = SUM(SOL(J),4);
1810 SUM(J) = SUM(J) + 2600;
1820 CALL OUTWAIT;
1830 OUTPUT (A) = SUM(J);
1840 END;
1850 CALL OUTWAIT;1860 OUTPUT (B) = SLASH;
1870 DO P = 0 TO 2;
1880 RETURN;
1890 IF P = 0 THEN DO;
1900 OFFLARE P BYTE;
1910 A(3) BYTE;
1920 GL DATA ('SOL=');
1930 OS DATA ('SSE');
1940 RI DATA ('IS=');
1950 MES DATA ('SWITCH VALVE CONTROLLER TO REMOTE');
1960 ***** SOLUTE LEVEL CHANGES *****
1970 ***** PREPARE CORE VALUE OF SOL FOR ILY *
1980 DO P = 0 TO 2;
1990 INPUT(27)=INIT13;
2000 DO P = 0 TO 2;
2010 CALL OUTWAIT;
2020 OUTPUT(8)=MES(P);
2030 END;
2040 CALL CHLF;
2050 CALL CHLF;
2060 SOL(2) = SOL(1);
2070 SOL(1) = SHR(SOL(1),4);
2080 SOL(2) = SHR(SOL(2),4);
2090 SOL(2) = SHR(SOL(2),4);
2100 DO P = 0 TO 2;
2110 SOL(P) = SOL(P) + 2600;
2120 END;
2130 FOR SOL: CALL CHLF;
2140 DO P = 0 TO 2;
2150 CALL OUTWAIT;
2160 OUTPUT (A) = OL(P);
2170 END;
2180 DO P = 0 TO 2;
2190 CALL OUTWAIT;
2200 OUTPUT (A) = SOL(P);
2210 END;
2220 CALL CHLF;
2230 CALL CHLF;
2240 DO P = 0 TO 2;
2250 CALL INWAIT;
2260 A(P) = INPUT (0);
2270 CALL OUTWAIT;
2280 OUTPUT (B) = A(P);
2290 IF A(P) = SLASH THEN GO TO SOL30K;
2300 END;
2310 DO P = 0 TO 2;

```

```

2320 SOL(P) = A(P);
2330 END;
2340 GO TO I0$SOL;
2350 /* RETURN FINAL VALUE OF SOL TO CORE FORMAT. */
2360 SOL$OK;
2370 DO P = 0 TO 2;
2380 SOL(P) = SOL(P) - 2000;
2390 END;
2400 SOL(1) = SHL(SOL(1),4);
2410 SOL(1) = SOL(1) + SOL(2);
2420 /* SOL IS NOW STORED IN BCD FORM IN SOL(0) AND SOL(1) */
2430 /* PREPARES SS FROM CORE FOR I/O */
2440 SS(1) = SS(0);
2450 SS(0) = SHR(SS(0),4);
2460 SS(1) = SHL(SS(1),4);
2470 SS(1) = SHR(SS(1),4);
2480 SS(0) = SS(0) + 2600;
2490 SS(1) = SS(1) + 2600;
2500 /*..... SOLUTE STEP CHANGES .....*/
2510 /* DIRECTION OF STEPPING MOTORS CHECKED */
2520 STEPL: FSR = INPUT (17);
2530 FSR = SHL (FSR,7);
2540 FSR = SHR (FSR,7);
2550 IF FSR = 1 THEN GO TO ADS;
2560 POLR = NEG;
2570 GO TO NXT1;
2580 ADS: POLR = POS;
2590 NXT1: CALL CRLF;
2600 CALL CRLF;
2610 DO P = 0 TO LAST(QS);
2620 CALL OUTWAIT;
2630 OUTPUT (8) = QS(P);
2640 END;
2650 /* I/O OF PREVIOUS VALUE */
2660 CALL OUTWAIT;
2670 OUTPUT (8) = POLR;
2680 DO P = 0 TO 1;
2690 CALL OUTWAIT;
2700 OUTPUT (8) = SS(P);
2710 END;
2720 CALL CRLF;
2730 CALL INWAIT;
2740 A(0) = INPUT (0);
2750 CALL OUTWAIT;
2760 OUTPUT (8) = A(0);
2770 IF A(0) = POLR THEN GO TO SC$OK; /* GOES TO SS INPUT */
2780 IF A(0) = SLASH THEN GO TO SS$OK; /* TO NEXT PARAMETER SETUP */
2790 OUTPUT (25) = 1; /* CHANGES STEPPING MOTOR DIRECTION */
2800 SC$OK;
2810 DO P = 0 TO 1;
2820 CALL INWAIT;
2830 SS(P) = INPUT (0);
2840 CALL OUTWAIT;
2850 OUTPUT (8) = SS(P);
2860 END;

```


*/

HOLDS TWO BCD DIGITS.

```

3420  THE EIGHT BIT BYTE. (ONE WORD), HOLDS TWO BCD DIGITS.
3430  C4= PROCEDURE
3440  1=0;
3450  1=DEF(0)/16+1000;
3460  1=DEF(0) AND 170+100;
3470  1=DEF(1)/16+100;
3480  1=DEF(1) AND 170;
3490  1=DEF(2)/16+100;
3500  1=DEF(2) AND 170;
3510  1=DEF(3)/16+100;
3520  1=DEF(3) AND 170;
3530  1=DEF(4)/16+100;
3540  1=DEF(4) AND 170;
3550  1=DEF(5)/16+100;
3560  1=DEF(5) AND 170;
3570  1=DEF(6)/16+100;
3580  1=DEF(6) AND 170;
3590  1=DEF(7)/16+100;
3600  1=DEF(7) AND 170;
3610  1=DEF(8)/16+100;
3620  1=DEF(8) AND 170;
3630  1=DEF(9)/16+100;
3640  1=DEF(9) AND 170;
3650  1=DEF(10)/16+100;
3660  1=DEF(10) AND 170;
3670  1=DEF(11)/16+100;
3680  1=DEF(11) AND 170;
3690  1=DEF(12)/16+100;
3700  1=DEF(12) AND 170;
3710  1=DEF(13)/16+100;
3720  1=DEF(13) AND 170;
3730  1=DEF(14)/16+100;
3740  1=DEF(14) AND 170;
3750  1=DEF(15)/16+100;
3760  1=DEF(15) AND 170;
3770  1=DEF(16)/16+100;
3780  1=DEF(16) AND 170;
3790  1=DEF(17)/16+100;
3800  1=DEF(17) AND 170;
3810  1=DEF(18)/16+100;
3820  1=DEF(18) AND 170;
3830  1=DEF(19)/16+100;
3840  1=DEF(19) AND 170;
3850  1=DEF(20)/16+100;
3860  1=DEF(20) AND 170;
3870  1=DEF(21)/16+100;
3880  1=DEF(21) AND 170;
3890  1=DEF(22)/16+100;
3900  1=DEF(22) AND 170;
3910  1=DEF(23)/16+100;
3920  1=DEF(23) AND 170;
3930  1=DEF(24)/16+100;
3940  1=DEF(24) AND 170;
3950  1=DEF(25)/16+100;
3960  1=DEF(25) AND 170;

```

```

3970      LOOP:      OUTPUT(20)=1;
3980      CALL QIRSEC;
3990      A(1)=INPUT(21);
4000      A(2)=INPUT(22);
4010      A(3)=INPUT(23);
4020      NH=(A(1) AND 170)+(A(1)/16)*10;
4030      NM=(A(2) AND 170)+(A(2)/16)*10;
4040      NS=(A(3) AND 170)+(A(3)/16)*10;
4050      IF NH<1VNH THEN GO TO LOOP;
4060      IF NM<1VNM THEN GO TO DONE;
4070      IF NS<1VNS THEN GO TO LOOP;
4080      IF NM>1VNM THEN GO TO LOOP;
4090      IF NS>1VNS THEN GO TO DONE;
4100      IF NS<1VNS THEN GO TO LOOP;
4110      RETURN;
4120      END WAITIO;
4130      ERROR:      PROCEDURE;
4140      DELTA=(PV+64)/128;
4150      IF LV>DELTA+PV THEN GO TO DONE;
4160      IF LV+DELTA<PV THEN GO TO DONE;
4170      STBL=1;
4180      RETURN;
4190      END ERROR;
4200      /* MARK SUBROUTINE.....*/
4210      /* THIS ROUTINE PRINTS THE PRESENT STATE OF THE PROGRAM
4220      /* ON THE TTY IN THE FORM "STATE = (FLAG) *". WHERE FLAG IS
4230      /* EITHER A, H, OR C. THE ACTUAL ASCII CHARACTERS ARE STORED
4240      /* IN THE LITERALS LABELED AS AA, BB, AND CC.
4250      MARK: PROCEDURE;
4260      DECLARE J BYTE,
4270      /* FLAG DATA ('STATE= ');
4280      DO J = 0 TO 3;
4290      CALL CHLF;
4300      END;
4310      DO J = 0 TO LAST(FLG);
4320      CALL OUTWAIT;
4330      OUTPUT (8) = FLG(J);
4340      END;
4350      CALL OUTWAIT;
4360      JUTPUT (8) = FLAG;
4370      DO J = 0 TO 1;
4380      CALL CHLF;
4390      END;
4400      RETURN;
4410      END MARK;
4420      /* ADDRESS SUBROUTINE.....*/
4430      /* IF THE PROGRAM IS ADSORBING, FSR WILL BE EQUAL TO ONE
4440      /* AND THE VALUE OF SS (GIVEN BY OPERATOR) WILL BE ADDED TO SOL.
4450      /* IF DESORBING, SS WILL BE SUBTRACTED FROM SOL.
4460      ADDRESS: PROCEDURE;
4470      DECLARE (A1,A2,B1,B2) BYTE;
4480      /* CHECK STEPPING MOTOR DIRECTION */
4490      FSR = INPUT (17);
4500      FSR = SHL(FSR,7);
4510      FSR = SHR(FSR,7);

```



```

5070 CALL WAIT10;
5080 END;
5090 STBLCHK=0;
5100 STBL=0;
5110 D1=(DEF(0)/16)*1000;
5120 D2=(DEF(0) AND 170)*100;
5130 D3=(DEF(1)/16)*10;
5140 D4=DEF(1) AND 170;
5150 LASTR=D1+D2+D3+D4;
5160 SSAGS=(LASTR+8)/16;
5170 IF STATE=0 THEN GO TO STATER;
5180 IF SOL(0)=0 THEN GO TO TEST;
5190 IF STATE=1 THEN GO TO STATEC;
5200 STATER: STATE=1;
5210 FLAG=88;
5220 CALL MARK;
5230 OUTPUT(26)=8832;
5240 OUTPUT(27)=8833;
5250 DO CTR=1 TO 25;
5260 CALL UIRSEC;
5270 END;
5280 OUTPUT(24)=1;
5290 CALL ADDRESS;
5300 IF SOL(0)>0 THEN FSR=0;
5310 OUTPUT(25)=1;
5320 GO TO START;
5330 TEST: IF SOL(1)>0 THEN GO TO NEXT;
5340 STATE=0;
5350 FLAG=AA;
5360 CALL MARK;
5370 OUTPUT(26)=AA32;
5380 OUTPUT(27)=AA33;
5390 HALT;
5400 STATEC: STATE=2;
5410 FLAG=CC;
5420 CALL MARK;
5430 OUTPUT(26)=CC32;
5440 OUTPUT(27)=CC33;
5450 GO TO START;
5460 EOF;

```

READY

APPENDIX B
CUT-BAR EXPERIMENT PLM PROGRAM

04MAY77 15:53:27 PTCARD20

```

640 VAL=VAL+TENX;
650 IF J=2 THEN GO TO RES;
660 J=J+1;
670 VAL=VAL+HMAX;
680 RETURN;
690 END VALIN;
700 VIDAROUT(0)=3200;
710 VIDAROUT(1)=2600;
720 VIDAROUT(2)=2600;
730 VIDAROUT(3)=2600;
740 CALL CHLF;
750 DO K=0 TO LAST(INITMES1);
760 CALL TTYOUTWAIT;
770 OUTPUT(0)=INITMES1(K);
780 END; CALL CHLF; CALL CRIF;
790 CALL VALIN;
800 MIN=VAL;
810 DO J=0 TO LAST(INITMES2);
820 CALL TTYOUTWAIT;
830 OUTPUT(0)=INITMES2(J);
840 END; CALL CHLF; CALL CRIF;
850 CALL VALIN;
860 CH=VAL;
870 TENS=CH/10;
880 ONES=CH-(TENS*10);
890 DO K=0 TO 3;
900 CALL SLOWWAIT;
910 OUTPUT(15)=VIDAROUT(K);
920 LNO;
930 DO K=0 TO 7;
940 CALL SLOWWAIT;
950 VIDARIN(K)=INPUT(6);
960 END;
970 DO K=0 TO 100;
980 CALL TIME(250);
990 END;
1000 IF VIDARIN(0)=2600 THEN GO TO RES;
1010 CALL CHLF; CALL CHLF; CALL CRIF;
1020 FIN=9;
1030 DO I=0 TO TENS;
1040 VIDAROUT(2)=I*2600;
1050 IF I=TERS THEN FIN=ONES;
1060 DO J=0 TO FIN;
1070 VIDAROUT(3)=J+2600;
1080 P1;
1090 DO K=0 TO 3;
1100 CALL SLOWWAIT;
1110 OUTPUT(15)=VIDAROUT(K);
1120 END;
1130 P2;
1140 DO K=0 TO 7;
1150 CALL SLOWWAIT;
1160 VIDARIN(K)=INPUT(6);
1170 END;
1180 P3;

```

```

1190      EXP=VIDARIN(0)-2600;
1200      VIDARIN(0)=2600;
1210      IF VIDARIN(1)=2550 THEN VIDARIN(1)=2400;
1220      P4:
1230      DO K=2 TO 7;
1240      IF VIDARIN(K)=2600 THEN VIDARIN(K)=2400;
1250      ELSE GO TO CONT;
1260      CONT:
1270      DO K=0 TO (11-EXP);
1280      PRINT(K)=VIDARIN(K);
1290      END;
1300      PRINT(11-EXP)=2560;
1310      DO K=(12-EXP) TO 0;
1320      PRINT(K)=VIDARIN(K-1);
1330      END;
1340      DO K=0 TO LAST(CHMES);
1350      CALL TTYOUTWAIT;
1360      OUTPUT(0)=CHMES(K);
1370      END;
1380      DO K=1 TO 3;
1390      CALL TTYOUTWAIT;
1400      OUTPUT(8)=VIDAROUT(K);
1410      END;
1420      DO K=0 TO 8;
1430      CALL TTYOUTWAIT;
1440      OUTPUT(8)=PRINT(K);
1450      END;
1460      DO K=0 TO LAST(MVMES);
1470      CALL TTYOUTWAIT;
1480      OUTPUT(8)=MVMES(K);
1490      END;
1500      CALL CRLF;
1510      END;
1520      DO K=1 TO (MIN-1);
1530      CALL MINUTE;
1540      END;
1550      GO TO LOOP;
1560      EUF

```

READY

APPENDIX C
DEVELOPMENT SYSTEM PDP-11 ASSEMBLY LANGUAGE PROGRAM

04MAY77 15:47:12 PICARD20

```

INC      R0
Cmp      R0,#400
BLT      LOOP3
BIS      #100,#177560
BIS      #100,#177564
.TTYIN
JMP      START
TYPE:    CLR      R3
LOOP4:   INC      R3
        Cmp      #4100,R3
        BGT      LOOP4
        MOVH     R2,#167772
        BIC      #400,#167772
        BIS      #400,#167772
        RTS      PC
MSG:     .ASCII   /TURN ON TAPE PUNCH/
        .BYTE    15,12
        .ASCIIZ   /AND STRIKE ANY KEY./
        .EVEN
B1:      .BYTE    0
B2:      .BYTE    0
B3:      .BYTE    0
B4:      .BYTE    0
DATA:    .BYTE    0
        .EVEN
TEMP:    .WORD    0
COL:     .WORD    0
LIST:    .BLKW    5
INBLK:   0
BUFF:    0
DEXT:    .RAD50   /LDA/
HAND:    .END     START
    
```

APPENDIX D
USERS MANUAL FOR THE DATA ACQUISITION IN THE LHMEL FACILITY

SECTION 1

PREVIEW

The Laser Hardened Materials Evaluation Laboratory (LHMEL) facility includes a dedicated Hewlett-Packard 9640A Multiprogramming System and a communication link to the SEL86 computer system. The OZ software system is a collection of programs residing both on the Hewlett-Packard system and the SEL86 system that acquires experimental data and performs real time analysis of data, permitting the LHMEL operator to rapidly alter experimental parameter values for best effect.

The experimental data that are acquired are:

- 1) Various experimental pre- and post-run operating parameter values (entered by the operator).
- 2) The run startup time.
- 3) The laser power (20 Hz sample rate).
- 4) The air temperature (20 Hz sample rate).
- 5) The static air temperature (20 Hz sample rate).
- 6) The total air pressure (20 Hz sample rate).
- 7) The time the laser shutter opens and closes.
- 8) The time the burn-through wire in front of the target is cut by the laser beam.
- 9) The time the burn-through wire behind the target is cut by the laser beam.

The principle component of the Hewlett-Packard system is a 21MX computer, operating under the Hewlett-Packard supplied RTE-C operating system. To begin an experimental run the operator uses RTE-C to activate the program OZ and enters the pre-run operating parameter values that have changed since the last experimental run. The operator then has the program OZGO activated to control the acquisition of the experimental data. Initialization of the laser control sequencer by the operator starts the program data acquisition sequence.

When the run terminates, the operator can use RTE-C to activate the program OZPST to enter the post-test operating parameter values. If desired, the operator can have the program DADMP activated to inspect the raw data acquired during the run. If the data are acceptable, the SEL86 program can be activated. Within this program the operator sends the raw data over the communication link to the SEL86 computer and initiates the execution of the analysis program SELOZ (on the SEL86). SELOZ computes the following:

- 1) The run time interval (time of shutter closure minus run startup time).
- 2) The shutter time interval (time of shutter closure minus time shutter opened).
- 3) The burn-through time interval (back burn-through wire time minus front burn-through wire time).
- 4) The laser power curve over the shutter time interval.
- 5) The laser power threshold time interval (the time when the laser power becomes less than, or equal to, the laser power threshold going down minus the time when the laser power becomes greater than, or equal to, the laser power threshold going up).
- 6) The maximum, minimum, and average values of laser power over the shutter time interval.
- 7) The maximum and average values of laser power over the laser power threshold time interval.
- 8) The total energy over the laser power threshold time interval.
- 9) The diameter and area of the beam on-target.
- 10) The laser power on-target curve over the shutter time interval.
- 11) The average power on-target over the shutter time interval.
- 12) The average power density on-target over the shutter time interval.
- 13) The energy density on-target over the shutter time interval.
- 14) The average power on-target over the burn-through time interval.
- 15) The average power density on-target over the burn-through time interval.
- 16) The energy density on-target over the shutter time interval.
- 17) The total temperature curve over the shutter time interval.
- 18) The minimum temperature over the shutter time interval.
- 19) The minimum temperature over the shutter time interval.
- 20) The RMS value of static air pressure over the shutter time interval.
- 21) The total air pressure curve over the shutter time interval.
- 22) The RMS value of total air pressure over the shutter time interval.
- 23) The Mach number.

The data and analytical results are conveniently formatted and printed on a high-speed printer at the SEL86 site. They are also archived on magnetic tape. Programs are provided to produce CALCOMP quality plots of the power on-target, total temperature, static pressure, and total pressure curves either in real-time or from the archived data. A summary of the analysis is printed for the operator on the LHMEI teletype. The operator can use this summary to select the experimental conditions for the next run.

SECTION 2

PROGRAM SUMMARIES

2.1 INTRODUCTION

There are five sets of routines within the OZ system:

- 1) RTE-C: the operating system on the 21MX computer.
- 2) The OZ programs: the set of 21MX resident routines that acquire the pre-run and post-run operating parameter values and experimental data.
- 3) SEL86: the 21MX resident routines that communicate with the SEL86 computer, sending data to it, starting programs on it, and receiving ASCII data from it for output on the LHMEI teletype.
- 4) SELOZ: the set of data analysis programs resident on the SEL86.
- 5) TPOUT: the program resident on the SEL86 that extracts data archived on magnetic tape and offers a plotting option.

Each of these sets of routines will be briefly described in the following sections. A more detailed description is found in Reference 15..

2.2 RTE-C

RTE-C is Hewlett-Packard's acronym for Real-Time Core-Based Software System. The procedure for initializing this system is found in Section 3.2. Once RTE-C is running, and any time it has control of the teletype, entering any character through the teletype will cause RTE-C to respond by typing an asterisk (*). At this time, the operator is free to enter any command described in the RTE-C operating manual.*

The operator is required to enter the time of day every time RTE-C is initialized. The format for this command is:

TM,day,h,min,sec

where day is a three-digit day-of-the-year (see Table 2-5 of the RTE-C operating manual), and h,min,sec is the current time on a 24-hour clock.

Hewlett-Packard Real-Time Core-Based Software System Programming and
Operation Manual, 1st Edition, 1970, Part No. 29101-93001.

Among the many functions of RTE-C is the control of the operator's access to the programs stored in the 21MX core. In order to execute a program, the operator types:

ON,pgnam

where pgnam is the name of the program to be executed.

Error messages from RTE-C are discussed on page 3-12 of the operating manual.

It should be noted that if a wrong character is typed "control-A" can be used to delete that character. RTE-C types a left-facing arrow to indicate that the character has been deleted. To delete an entire line, type "rubout". RTE-C types a back slash to indicate that the line has been deleted.

2.3 THE OZ ROUTINES

2.3.1 Introduction

The following OZ routines can be called using the RTE-C ON,pgnam command:

OZ
OZGO
DADMP
OZPST.

In addition, two other routines are called internally by RTE-C:

OZAD
OZUNI.

2.3.2 OZ

The OZ routine permits entry of pre-run operating parameter values. When the program is started by RTE-C, this heading is typed on the LHMELE teletype:

OZ, VERSION xx/xx/xx.

After this, the system waits for the operator to enter a two-letter code corresponding to a pre-run operating parameter. (These codes

are listed in Table I.) The code should be followed by a carriage return.

OZ searches its internal table of codes. If it finds a match it types:

ENTER XX

where XX is an echo of the operator-supplied code. A carriage return is executed and OZ waits for the operator to enter the appropriate parameter value followed by a carriage return.

If no match is found, OZ types:

NO MATCH

followed by a carriage return and waits for the code to be reentered.

Any number of parameters and their values can be entered during an OZ execution, and their order is arbitrary. This permits entry of only those parameters that have changed value since the previous run.

If a parameter value is entered incorrectly, the parameter code and value can be reentered.

All numeric parameter values should be entered in F7.3 format (i.e., in the form 999.999). For alphanumeric parameter values, the length of the field varies with parameters as defined in Table I.

To notify OZ that no more parameters are to be entered, type XX for the parameter code. OZ types:

END OZ

and returns control to RTE-C.

2.3.3 OZGO

The OZGO routine initiates the data acquisition process. Upon entering the routine, all the flags and pointers are initialized and the rear burn-through wire clock is zeroed.* The following message is printed on the teletype:

*Upon completion of a run a zero rear burn-through wire time indicates that the target did not burn through.

TABLE I
PRE-RUN OPERATING PARAMETERS

PARAMETER	CODE	FORMAT	UNITS
sample id	SI	A6	
test number	TN	A4	
distance between mirror and sample	DI	F7.3	cm
HE dome pressure	HD	F7.3	psig
N2 dome pressure	ND	F7.3	psig
CO2 dome pressure	CD	F7.3	psig
laser pressure	LA	F7.3	mm Hg
exhaust pressure	EP	F7.3	mm Hg
high voltage	KV	F7.3	kilovolts
run time	RT	F7.3	sec
HE line pressure	HL	F7.3	psig
N2 line pressure	NL	F7.3	psig
CO2 line pressure	CL	F7.3	psig
WT back pressure	WB	F7.3	psig
WT line pressure	WL	F7.3	psig
WT plenum temperature	WP	F7.3	deg
WT nozzle design, x dim	WX	F7.3	cm
WT nozzle design, y dim	WY	F7.3	cm
initial load	IL	F7.3	kilopounds

READY..AUTO?

The operator should respond by entering a "Y" or an "N" followed by a carriage return.

If a "Y" is entered, the system waits up to 20-seconds for a signal from the laser control system (on channel 0 of the universal interface card data buffer).

(a) If the pulse does not occur within 20-seconds, this message is typed on the teletype:

TIME UP

and control is returned to RTE-C.

(b) If the pulse does occur within the time limit, the OZUNI routine sets the experiment initialize flag IF4 and records the time of the pulse as the "start-up" time. Control is returned to RTE-C.

If an "N" is entered, the experiment initialize flag IF4 is set, the current time is recorded as the "start-up" time, and OZAD and OZUNI are started. Control is returned to RTE-C.

2.3.4 DADMP

The DADMP routine generates a teletype listing of the values of pre-run and post-run parameters, the flags, the pointers, the data arrays, and the time buffers.

The numbers under the "FLAGS AND POINTERS" banner are the values of the flags and pointers in this sequence:

IF4- experiment initialize flag; incremented each time the experiment is initialized during a run.

IF0- front burn-through wire flag; incremented each time a front burn-through wire signal is detected during a run.

IF1- rear burn-through wire flag; incremented each time a rear burn-through wire signal is detected during a run.

IF2- shutter open flag; incremented each time a shutter open signal is detected during a run.

IF3- shutter close flag; incremented each time a shutter close signal is detected during a run.

IFER - previous universal interface data register contents (decimal equivalent).
IRBUF- current universal interface data register contents (octal value).
IFIN - set when flags initialized and reset when run completed.
IPNTR- the number of filled entries in the data arrays (decimal value).

The sequence of numbers labeled "START UP=" down through "SHUTTER CLOSE=" are associated with the clock buffer entries. The order of these entries on a single line is: tens of msec, seconds, minutes, hours, days (all in decimal).

The values in the data arrays are in units of volts.

2.3.5 OZPST

The OZPST routine is used to enter the post-run operating parameter values. When the program is started by RTE-C, the following heading is printed on the LHMEI teletype:

OZPST,VERSION XX/XX/XX.

At this point the system idles, waiting for the operator to enter one of the two-letter codes corresponding to a post-run operating parameter listed in Table II. The code should be followed by a carriage return.

OZPST searches its internal table of codes. If a match is found, it types

ENTER XX

where XX is the echo of the operator-supplied code. A carriage return is executed and OZPST waits for the operator to enter the appropriate parameter value followed by a carriage return.

If no match is found, OZPST types

NO MATCH

followed by a carriage return and waits for the code to be reentered.

All the parameter values are numeric and should be entered in an F7.3 format (i.e., in the form 999.999).

TABLE II
POST-RUN OPERATING PARAMETERS

PARAMETER	CODE	FORMAT	UNITS
HE upstream pressure	HU	F7.3	psig
N2 upstream pressure	NU	F7.3	psig
CO2 upstream pressure	CU	F7.3	psig
laser pressure	LP	F7.3	mm Hg
exhaust pressure	EX	F7.3	mm Hg
current	CR	F7.3	milliamps
WT exhaust pressure	WE	F7.3	psig

To notify OZPST that no more parameters are to be entered, the operator should type "XX" for the parameter code. OZPST types:

END OZPST

and returns control to RTE-C.

2.3.6 OZAD

The purpose of OZAD is to control the A/D converter.

When OZAD is started by OZUNI, it checks the IF4 flag (experiment initialize) to determine if the run has been initialized by the OZGO routine. If it has not, this message is typed on the teletype:

OZAD NOT INIT.

Control is returned to RTE-C.

If the run has been initialized, the IF3 flag (shutter closed) is examined to determine if the run has been completed. If it has, the message

A/D DONE

is typed on the teletype and control is returned to RTE-C. If the run is still in progress, a single reading is taken on each of the first four channels (CH0-CH3) of the A/D converter. These readings are stored in the next available location of the four arrays PMETR(CH0), TTEMP(CH1), SPRES(CH2), and TPRES(CH3). Then the pointer used to indicate the next available location, IPNTR, is incremented and that value compared with the maximum dimension of the four arrays (101). If the arrays are not full, OZAD reschedules itself to execute 50 msec later and control is returned to RTE-C.

If the arrays are full, which occurs five seconds after the run begins, this message is printed on the teletype:

ARRAYS FULL

A/D DONE

Control is returned to RTE-C.

2.3.7 OZUNI

OZUNI is the program that processes the signals input to the Universal Interface Card (12930A). It is started by the OZGO program and it reschedules itself to run every 20 msec until either the run is completed, an error has occurred, or 20 seconds has passed without a "start-up" signal. This program should not be called by the operator using the RTE-C "ON,OZUNI" command.

The Universal Interface Card data channel input register is assumed by OZUNI to be wired as follows:

- bit 0: run start-up signal wire
- bit 1: shutter signal wire
- bit 2: front burn-through signal wire
- bit 3: back burn-through signal wire.

All signals are assumed to be single-ended.

The logical meaning of each of the signals is assumed to be as follows:

Run start-up signal: the occurrence of the 0 to 5v transition indicates start of run. The signal remains at 5v for the duration of the run.

Shutter signal wire: the occurrence of the first 0 to 5v transition after the start of a run indicates the opening of the shutter. Occurrence of the next 5v to 0v transition indicates the closing of the shutter. The signal remains at 5v in between.

Front burn-through signal: the occurrence of the first 0 to 5v transition after the start up of a run indicates that the front burn-through wire has burned through. The signal remains at 5v for the duration of the run.

Back burn-through signal: the occurrence of the first 0 to 5v transition after the start of a run indicates that the back burn-through wire has burned through. The signal remains at 5v for the duration of the run.

Upon the first entry into the program during a run, the FINIT flag is examined to determine if the OZGO program initiated OZUNI. If not, the program

EARLY UNIV INT

is typed on the teletype and control is returned to RTE-C. This prevents the operator from starting OZUNI using the RTE-C "ON,OZUNI" command. Otherwise, each time OZUNI runs, the first thing it does is load the Universal Interface Card's data register into the A register, where it is exclusive-ored with the contents of the data register obtained the last time OZUNI ran. The resulting word has a bit set in every position where the level of the data register has changed since the last time OZUNI ran. This word is processed as shown in the accompanying flowchart in Figure 1. During the course of processing, if a logical error occurs, (e.g., two-level changes in the front burn-through wire signal during the course of one run), the message "ERROR IN UNIV INT" is printed on the teletype and data acquisition for the run is aborted. The source of the error can be determined by running DADMP and examining the values for IFER and IRBUF. If no logical error occurs, OZUNI checks the shutter-closed flag to determine if the shutter-closed event has occurred. If it has, the program terminates. If it hasn't the program reschedules itself to run 20 msec later.

After OZGO turns OZUNI on, OZUNI runs every 20 msec, checking for the start-up signal. OZGO times the event. If the signal is not generated in 20 seconds, OZGO notifies OZUNI through the IFIN flag, and OZUNI does not reschedule itself.

2.4 SEL86

The SEL86 routine controls the user's communication with the SEL 86 computer. SEL86 is brought up by typing:

ON,SEL86.

The response of the terminal is

-SEL TERMINAL SUPPORT SYSTEM, TERMINAL TYPE -
ENTER FUNCTION CODE, ?, OR CR TO TERMINATE:

The informed user will recognize this as the normal mode of entry into the SEL 86 Terminal Support System (TSS). All the options

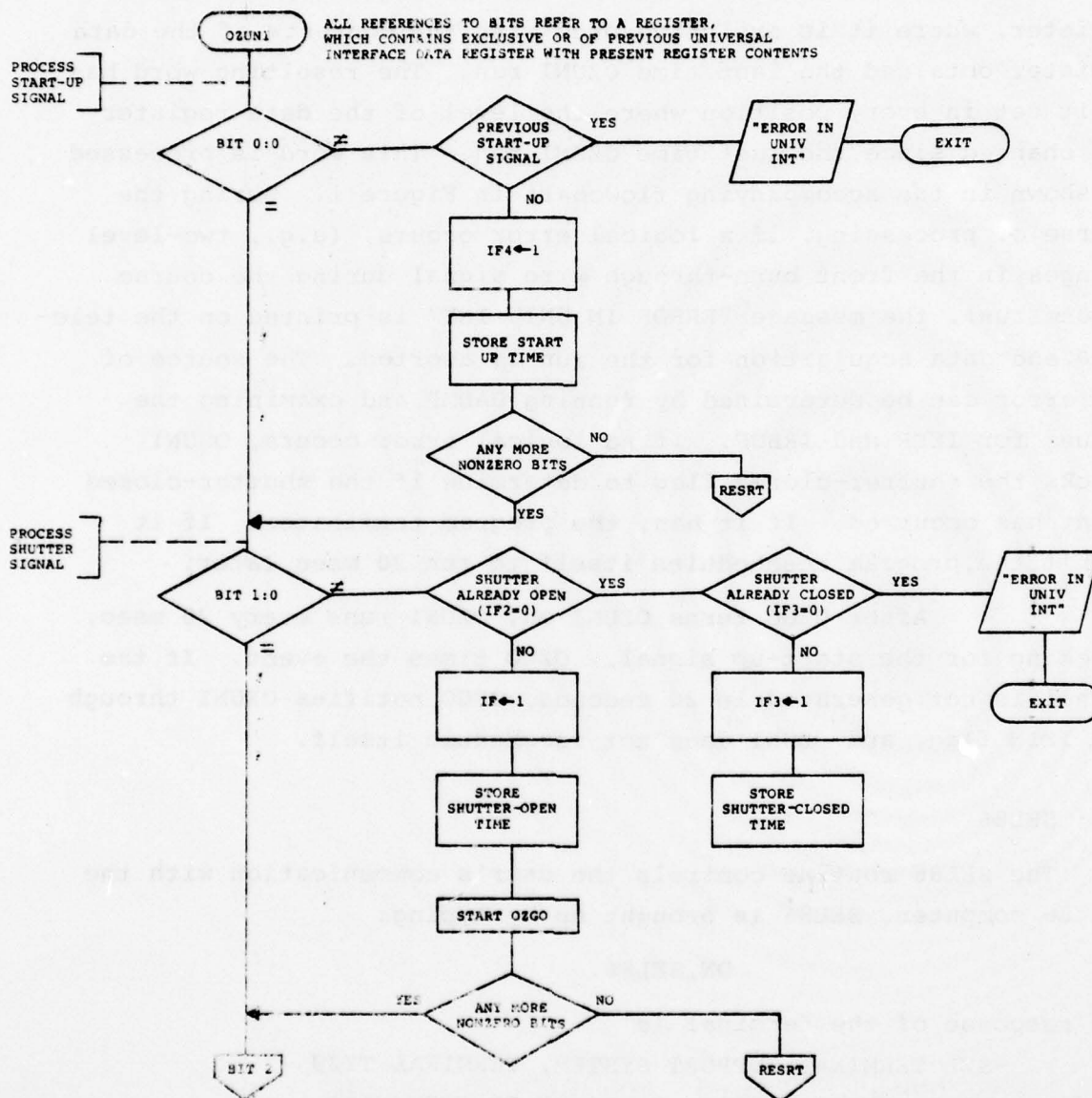


Figure 1. Flowchart of OZUNI.

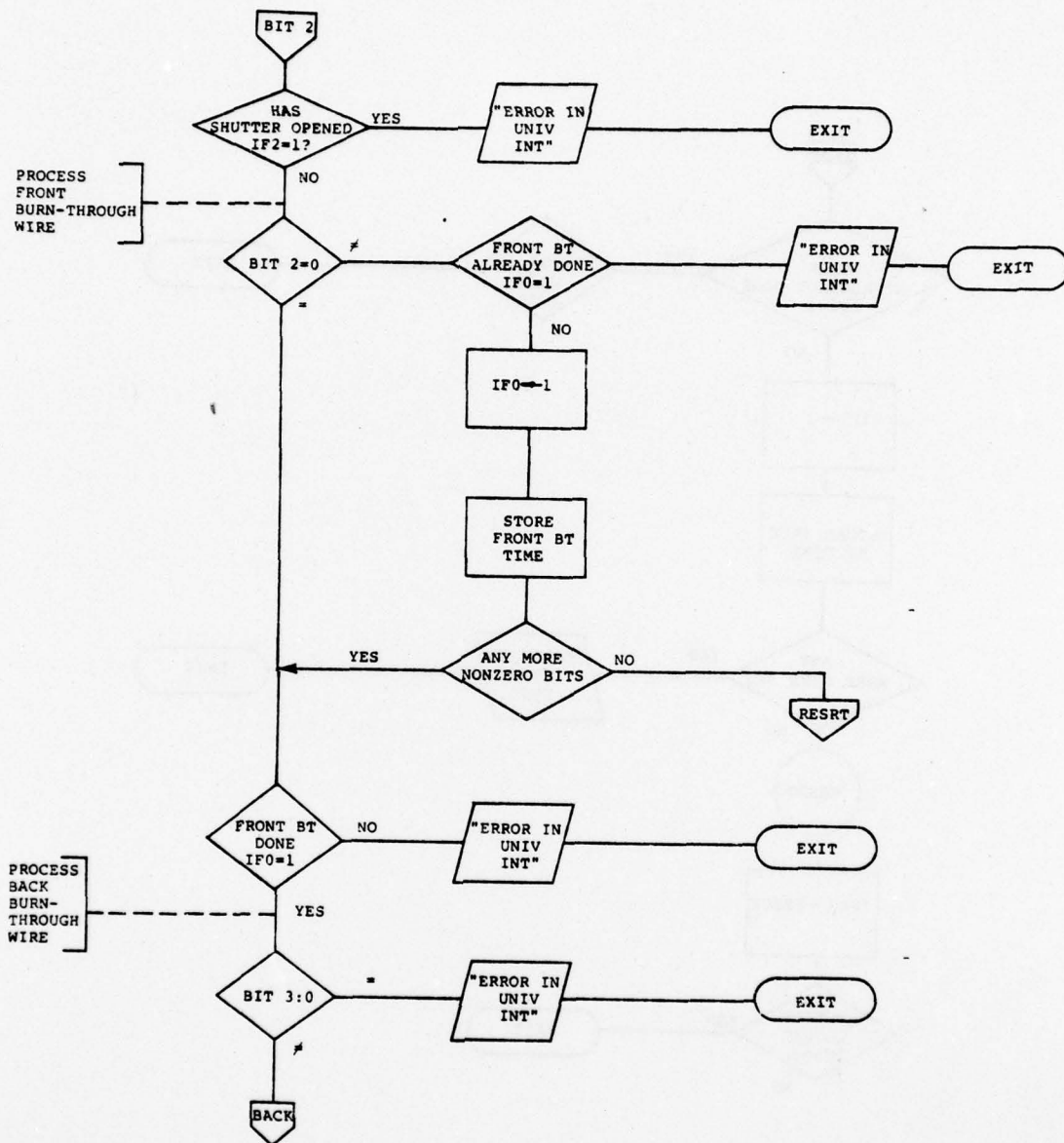


Figure 1. Flowchart of OZUNI. (Continued)

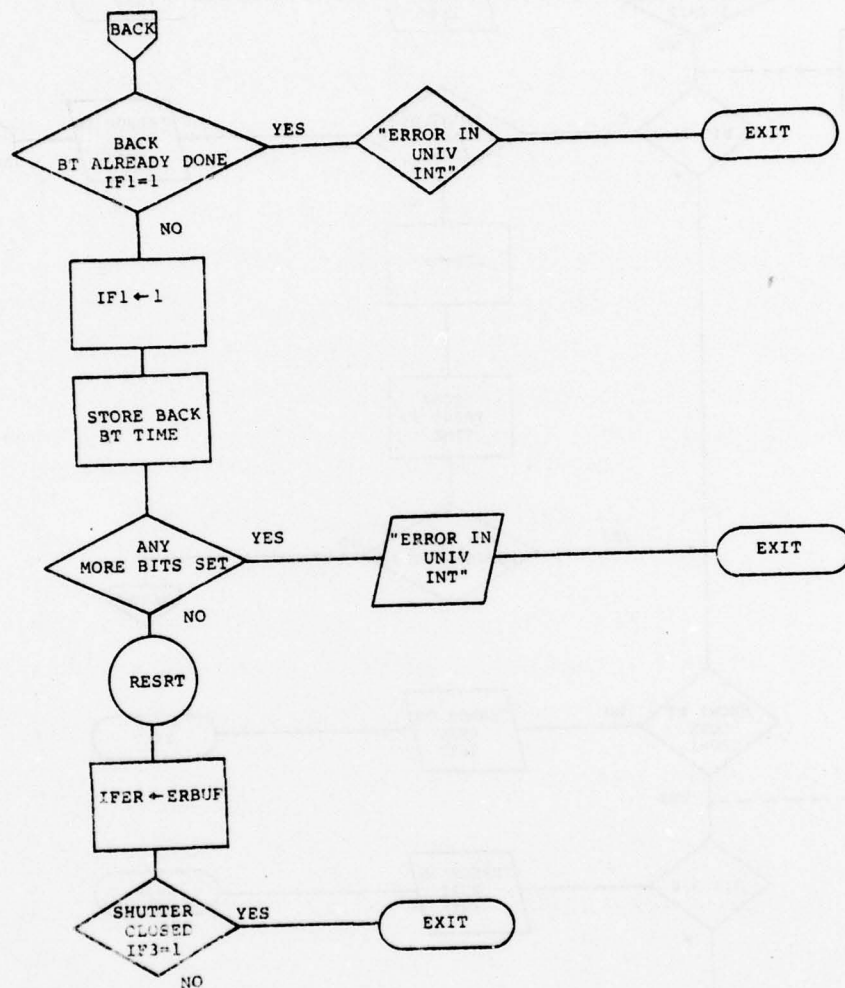


Figure 1. Flowchart of OZUNI. (Continued)

of this system are now open to the user. The normal protocol for an experiment is to use TSS to transfer the COMMON area of RTE-C to a disk file on the SEL 86 and then to run SELOZ.

To transfer the COMMON area, type "RM" in response to the above request:

ENTER FUNCTION CODE,?, OR CR TO TERMINATE: RM

The terminal will respond with a question mark and the user should type Y:

?Y

The terminal will follow this with another

ENTER FUNCTION CODE,?, OR CR TO TERMINATE:

The user should respond with:

ENTER FUNCTION CODE,?, OR CR TO TERMINATE: PM.

The terminal will type:

- BEGIN PROGRAM MONITOR -
PROGRAM (, #SECONDS):

The proper user's response is:

PROGRAM (, #SECONDS): MØSELOZ,200.

This will start the execution of SELOZ.

When SELOZ has completed execution, the normal sequence of interaction is

PROGRAM(, #SECONDS): control-D
ENTER FUNCTION CODE,?, OR CR TO TERMINATE: control-D
TERMINAL TYØØ LOGGED OUT

At this point control is returned to RTE-C.

At the end of a day's runs, provide the following command to RTE-C:

*OF,SEL86,1.

A more technical description of SEL86 is found in the Programmer Comments section of Reference 15.

AD-A058 193

DAYTON UNIV OHIO RESEARCH INST
AFML SCIENTIFIC AND ENGINEERING COMPUTER SUPPORT.(U)

F/G 9/2

UNCLASSIFIED

2 OF 2

AD
A058193

DEC 77 D L FORD, T WOOD, M DENNIS, D G LEET
UDRI-TR-77-10

AFML-TR-77-231

F33615-75-C-5191

NL



2.5 SELOZ

SELOZ and its supporting routines analyze the data obtained by OZ, generate summaries of the analysis produce plots, and store the data on digital magnetic tape.

When SELOZ is initiated through SEL86 it performs the following procedure:

- 1) Send the message "BEGINNING SELOZ" to the LHMEI teletype.
- 2) Input the Calibration Deck (MØ\$CALDK).
- 3) Input the experimental data (MØ\$OZFIL).
- 4) Calculate the time the shutter was open (TS):

$$TS = \text{shutter close time (ISHC)} - \text{shutter open time (ISHO)}.$$

- 5) Calculate the time required to burn through the sample (TB):

$$TB = \text{back burn through wire time (IBTB)} - \text{front burn-through wire time (IBTF)}.$$

- 6) Calculate the time between the shutter open time and the time the front burn-through wire burned through (TD):

$$TD = \text{front burn-through wire time (IBTF)} - \text{shutter open time (ISHO)}.$$

- 7) If TD is greater than 30 seconds or negative, dump results, which will consist of zero values, and exit.
- 8) Convert the power meter readings to floating point volts:

$$CHØ(I) = \text{FLOAT(IAND(IMETR(I), B177760))} * .0003125 .$$

(Note: This procedure is documented in the HP 91000A A/D Converter Manual.)

- 9) Convert power meter volts to laser power (PL(I)):

$$PL(I) = \text{power volts} + (\text{power responsivity/beam splitter reflectivity})$$

or

$$PL(I) = CHØ(I) + (C(2)/C(1))$$

(See Table III for Calibration Table parameter definitions as stored in the program's arrays.)

- 10) Convert laser power to laser power on target (PT(I)):

$$PT(I) = (1 - \text{beam splitter reflectivity}) * (\text{combined reflectivity of 3 mirrors}) * (\text{laser power})$$

or

$$PT(I) = (1.-C(L)) * C(3) * PL(I)$$

- 11) Find the time (TL1) at which laser power (PL) first equals or exceeds the laser power threshold (C(4)).
- 12) Find the first time (TL2) after TL1 at which the laser power (PL) decreases below the laser power threshold (C(4)).
- 13) Calculate the difference between TL1 and TL2 (TL):

$$TL = TL2 - TL1 .$$

- 14) Define the TL time interval as the time interval from TL1 to TL2 inclusive.

IBINL1 is the element in any of the four data arrays corresponding to TL1 and IBINL2 is the element corresponding to TL2. Then $NL = IBINL2 - IBINL1 + 1$ is the number of elements in any array over the TL time interval. Calculate the average laser power over the TL time interval (PLLA):

$$PLLA = \frac{1}{NL} \sum_{I=IBINL1}^{IBINL2} PL(I) .$$

- 15) Calculate the maximum laser power (PLLMX) over the TL time interval:

$$PLLMX = \max\{PL(I) \mid I=IBINL1, IBINL2\} .$$

- 16) Define the TS time interval as the time interval from shutter open time (ISHO) to shutter close time (ISHC) inclusive. Define IBINS1 as the element in any of the four data arrays corresponding to ISHO and IBINS2 as the element corresponding to ISHC. $NS = IBINS2 - IBINS1 + 1$ is the number of elements in any data array over the TS time interval. Calculate the average laser power over the TS time interval (PLSA):

$$PLSA = \frac{1}{NS} \sum_{I=IBINS1}^{IBINS2} PL(I) .$$

- 17) Calculate the maximum laser power over the TS time interval (PLSMX):

$$PLSMX = \max \{PL(I) | I=IBINS1, IBINS2\} .$$

- 18) Calculate the minimum laser power over the TS time interval (PLSMN):

$$PLSMN = \min \{PL(I) | I=IBINS1, IBINS2\} .$$

- 19) Calculate the integral of laser power over the TL time interval (WJ):

$$WJ = .050 * \sum_{I=IBINL1+1}^{IBINL2} PL(I) .$$

The value .050 is in units of seconds and is the bin width.

- 20) Calculate the average laser power on target over the TS time interval (PTSA):

$$PTSA = \frac{1}{NS} \sum_{I=IBINS1}^{IBINS2} PT(I) .$$

- 21) Define the TB time interval as the interval from the time the front burn-through wire burned through to the time the back burn through wire burned through. Define IBINB1 as the element in any of the four data arrays corresponding to IBTF and IBINB2 as the element corresponding to IBTB. NB - IBINB2-IBINB1 is the number of elements in any data array over the TB time interval. Calculate the average laser power on target during the TB time interval (PTBA):

$$PTBA = \frac{1}{NB} \sum_{I=IBINB1}^{IBINB2} PT(I) .$$

- 22) Compare the mirror-to-sample length supplied in the experimental data (FID(1)) with the three sets of distance bounds in the Calibration Table. Select the correct slope (C7) and intercept value (C6) and compute the area of the laser beam

at the target:

$$\text{DIAM} = \text{C6} + \text{C7} * \text{FID}(1)$$

$$\text{AREA} = \left(\frac{\pi}{4}\right) * \text{DIAM}^2$$

- 23) Calculate the average laser power on target per unit area of the target over the TS time interval (PDSA):

$$\text{PDSA} = \text{PTSA} / \text{AREA}$$

- 24) Calculate the average laser power on target per unit area of the target over the TB time interval:

$$\text{PDBA} = \text{PTBA} / \text{AREA}$$

- 25) Calculate the integral of laser power on target over the TS time interval (EDSA):

$$\text{EDSA} = .050 * \sum_{\text{I}=\text{IBINS1}+1}^{\text{IBINS2}} \text{PT}(\text{I})$$

- 26) Calculate the integral of laser power on target over the TB time interval (EDBA):

$$\text{EDBA} = .050 * \sum_{\text{I}=\text{IBINB1}+1}^{\text{IBINB2}} \text{PT}(\text{I})$$

- 27) Convert the air temperature readings to floating point volts:

$$\text{CH1}(\text{I}) = \text{FLOAT}(\text{IAND}(\text{ITEMP}(\text{I}), \text{B177760})) * .0003125$$

- 28) Convert air temperature volts to degrees

$$\begin{aligned} \text{air temperature, degrees} &= (\text{air temperature, volts}) + \\ &\quad \text{air temperature channel responsivity, degrees/volt} \end{aligned}$$

or

$$\text{CT}(\text{I}) = \text{CH1}(\text{I}) * \text{C}(12)$$

- 29) Convert the static pressure readings to floating point volts:

$$\text{CH2}(\text{I}) = \text{FLOAT}(\text{IAND}(\text{IPRES}(\text{I}), \text{B177760})) * .0003125$$

30) Convert static pressure volts to psi:

$$\text{static pressure, psi} = (\text{static pressure, volts}) * \\ (\text{static pressure channel responsivity, psi/volt})$$

or

$$CT2(I) = CH2(I) * C(13) .$$

31) Convert the total pressure readings to floating point volts:

$$CH3(I) = \text{FLOAT}(\text{IAND}(\text{JPRES}(I), B177760)) * .0003125 .$$

32) Convert total pressure volts to psi:

$$\text{total pressure, psi} = (\text{total pressure, volts}) * \\ (\text{total pressure channel responsivity, psi/volt})$$

or

$$CT3(I) = CH3(I) * C(14) .$$

33) Compute the minimum air temperature over the TS time interval (CTMN):

$$CTMN = \text{Min} \{CT(I) \mid I=IBINS1, IBINS2\} .$$

34) Compute the RMS value of the static pressure over the TS time interval (PSR):

$$PSR = \sqrt{\frac{1}{NS} \sum_{I=IBINS1}^{IBINS2} CT2(I)^2} .$$

35) Calculate the RMS value of total pressure over the TS time interval (PTR):

$$PTR = \sqrt{\frac{1}{NS} \sum_{I=IBINS1}^{IBINS2} CT3(I)^2} .$$

36) Calculate the mach number (CMACH):

$$CMACH = \sqrt{5 * [(\frac{PTR}{PSR})^{.285714} - 1.]} .$$

- 37) Print the Calibration Table.
- 38) Print the evaluation sheets and dumps of various arrays.
- 39) Send the post-test summary to the LHMEI teletype.
- 40) Send "SAVE DATA?" message to LHMEI teletype. Wait for an answer.
If the answer is not "N", sent the message "NEW TAPE?" to the LHMEI teletype. If the response to "NEW TAPE?" is "N", send the response to "FIRST RUN?" to the LHMEI teletype. If the response to "FIRST RUN?" is "N", then this is not the first run of the day and the run data are added to the tape. If the response to "FIRST RUN?" is anything else, then a search of the magnetic tape is made for the last run on the tape before the run data are added to the tape. If the response to "NEW TAPE?" is anything other than "N" then the tape is assumed to be empty and the run data is written without checking whether or not the tape is at the end of useful information.
- 41) Send "PLOTS?" message to LHMEI teletype. Wait for an answer.
If the answer is not "N", generate the plots.

TABLE III
CALIBRATION TABLE PARAMETERS IN ARRAYS

ARRAY LABEL	CONTENTS
DATE	calibration table date
C(1)	beamsplitter reflectivity
C(2)	power responsivity (watts/v)
C(3)	combined reflectivity of 3 mirrors
C(4)	laser power threshold (watts
C(5)	mirror focal length (m)
D1(1)	lower mirror distance bound (cm) } diameter parameters used
D2(1)	
C(6)	slope
C(7)	intercept
D1(2)	lower mirror distance bound (cm) } diameter parameters used
D2(2)	
C(8)	slope
C(9)	intercept
D1(3)	lower mirror distance bound (cm) } diameter parameters used
D2(3)	
C(10)	slope
C(11)	intercept
C(12)	air temperature channel responsivity (degree/volt)
C(13)	static pressure channel responsivity (psi/volt)
C(14)	total pressure channel responsivity (psi/volt)

SECTION 3
STANDARD OPERATING PROCEDURES

3.1 BASIC PROCEDURE

1. Before each run in which the tape drive or plotter is to be used, or before each series of runs if the runs are going to follow one after another, call the SEL operator (Cindy) at 53778 and inform her of the equipment requirements. Also inform her whether or not the tape is to be rewound at the end of the run.
2. Start RTE-C using Section 3.2.
3. Load the OZUNI program using the RTE-C "LO" command.
4. If necessary, modify the Calibration Table using the procedure in Section 3.3.
5. Bring up OZ to set the pre-run parameters values.
6. Bring up OZGO for the experimental run.
7. Bring up OZPST to set the post-run parameter values.
8. If desired, bring up DADMP to examine the data.
9. Bring up SEL86 and transfer the data to the OZFIL file in the SEL.
10. Run SELOZ.
11. Bring up OZ to change any pre-run parameters that are to be changed for the next run.
12. Go to step 6 for another run.

3.2 STARTING RTE-C

1. Press Master Power button on upper right side of 9640A rack.
2. Insert Key into front panel of 2lMX. Turn key counter-clockwise to R and then back to operate.
3. Turn on paper tape reader.
4. Turn on teletype (to line position).
5. Load RTE-C paper tape into paper tape reader.
6. Enter 001500₈ into switch (5) register.
7. Press IBL.

8. Press RESET.
9. Press RUN.
10. When the paper tape has been read, the light pattern on the front panel should be 100277 and this should be the contents of the S register.
11. Enter 000002₈ into P register.
12. Press Preset and RUN.

3.3 MODIFYING THE CALIBRATION TABLE

The Calibration Deck used by the SELOZ program can be changed from the LHMEL teletype using the SEL86 TEST EDITOR. Any time after RTE-C has been started, an

ON,SEL86

command will initialize communication with the SEL. The following sequence will provide access to the MØ\$CALDK Edit file, which can then be changed using the EDITOR commands:

```
-SEL TERMINAL SUPPORT SYSTEM, TERMINAL TYØØ
ENTER FUNCTION CODE,?, OR CR TO TERMINATE: ED

-BEGIN TEST EDITOR.
ENTER 2-CHARACTER USER CODE OR CR TO TERMINATE: MØ
COMMAND? USE CALDK
COMMAND?
```

When the changes have been made, the following response saves the new file:

```
COMMAND? SAVE CALDK SCR
```

A response of EXIT terminates the EDITOR.

The following command terminates SEL86:

```
OF,SEL86,1 .
```

APPENDIX E
FORTRAN PROGRAM SER TO CALCULATE RESPONSE FUNCTIONS

[illegible]

```

150 CONTINUE
161 I=1
162 J=2
163 THK(TIME)=S0
CARR CALCULATE P1
DO 140 J=1,NOL
S0=S0+H(J)/H01(J)/CPI(J)/SUM**2*DEL(J)+S05
P1(TIME)=S05*H(J)/SUM
P03(TIME)=CUMV(J)+ORP(J)-RAU(J)
CARR CALCULATE P5
P5(TIME)=OLLOSS(J)/P03(TIME)
CARR CALCULATE THE AVERALS OF P4,P5,P6
CARR INFORMATION OVER X
DO 150 J=1,NOL
S02=P04(J)*DEL(J)+S02
S03=S03+H(J)*SUM*MDG(J)/RR(J)*DEL(J)
CARR -503. IS THE HEAT OF FORMATION OF THE PLASTIC (BASELINE)
S04=S04+MDG(J)/RR(J)*(-503)*DEL(J)
WE=WT+DEL(J)*H01(J)*.001/12.
150 CONTINUE
S02=S02/SUM4
P4(TIME)=S02/P03(TIME)
S03=S03/SUM4
P5(TIME)=S03/P03(TIME)
S04=S04/SUM4
P6(TIME)=S04/P03(TIME)
WE(TIME)=WT
160 CONTINUE
170 CONTINUE
180 CONTINUE
181 I=1
182 J=1
183 S2=S3=S4=S5=S6=0.
CARR CALCULATE THE INTEGRALS OF P4,P5,P6
CARR FOR EACH TIME INTERVAL
PRINTA," TIME P1 P2 P3 P4 P5 P6 WEIGHT"
DO 190 J=1,IM1
J1=J+1
DIH=(H(J1)-H(J))/H0.5
P2(J)=DIH*P1(J)+DIH*S2
S2=P1(J)
P04(J)=DIH*(P4(J)+S4)
S4=P4(J)
P05(J)=DIH*(P5(J)+S5)
S5=P5(J)
P06(J)=DIH*(P6(J)+S6)
S6=P6(J)
190 CONTINUE
CARR SUM OVER ALL INTERVALS AND OUTPUT DATA
S2=S3=S4=S5=S6=0.
DO 200 J=1,IM1
J1=J+1
P2(J)=P2(J)+S2
S2=P2(J)
P04(J)=P04(J)+S4
S4=P04(J)
P05(J)=P05(J)+S5
S5=P05(J)
P06(J)=P06(J)+S6
S6=P06(J)
PRINT 260, H(J1),P1(J),P2(J),P3(J),P4(J),P5(J),P6(J),WE(TJ)
CARR WRITE OUTPUT FOR PUNCH
WRITE (10,2/0) H(J1),TS(J1),P1(J),P2(J),WE(TJ)
200 CONTINUE
201 J=J+1

```


DATE 0004 1111 END 05 151 1111

● ● ● ● ● ● ●

102

APPENDIX F
FORTRAN PROGRAMS FOR THE ANALYSIS OF MICROSTRUCTURES

```

1  C-
5  C-PROGRAM PARSEP1 DETERMINES THE PARTICLES IN THE THRESHOLDED IMAGE AFTER THE
   C-INTERIOR PIXEL VALUES HAVE BEEN GENERATED BY PROGRAM PARSEP. IN ITS PRESENT
   C-STATE IT DETERMINES THE PARTICLE CORES, UPPER EXTREMES, AND LOWER EXTREMES.
   C-CONVERTS THEM TO INTERVAL NOTATION, AND OUTPUTS THE PARTICLES (IN INTERVAL
   C-NOTATION) ON TAPE5.
   C-
10  DIMENSION R(3),C(32),O(1500)
15  DIMENSION STACK(1000,3)
   INTEGER R,C,O
   INTEGER STACK,TOP
   TOP=1
   READ 1000,NROWS,NCOLS
   FORMAT(2I5)
   IFLAG=0
   IV=3
   IV1=4
   NWR=NCOLS+2
   C-
20  C-INITIALIZE I POINTER ARRAY (B) AND SHIFT ARRAY (C)
   C-
25  R(1)=1
   B(1)=NWR+1
   B(3)=B(2)+NWR
   C(32)=58
   C(1)=0
   KE=60
   DO 10 I=2,31
     KE=KE-2
     C(I)=KE
   10  K=1
   PRINT 5000,C
   PRINT 5000,B
   FORMAT(1H1,3I4)
   K=3
   CALL FIRST(O,B,C,NWR,K,IV,I,J,II,IROW,IV1)
   CALL PARFIND(O,B,C,I,J,II,IROW,STACK,TOP,K,NROWS,IV,IV1,NWR)
   CALL STKSRT(STACK,TOP)
   L=1
   CALL INTNOTN(STACK,TOP,L,LL)
   CALL SURHD2SD,R,C,STACK,TOP,L,NROWS,NWR,IV,IV1,IROW)
   CALL TEST(O,B,C,STACK,TOP,L,NROWS,NWR,IV,IV1,IROW)
   DO 50 I=1,25
     PRINT *,(STACK(I,J),J=1,3)
   50  STOP
   100 FORMAT(1H *,FIRST INFO *,3(I5,5X))
      END

```

```

1      SUBROUTINE FIRST(I,J,K,L,M,N,P,Q,R,S,T,U,V,W,X,Y,Z,AA,AB,AC,AD,AE,AF,AG,AH,
      DIMENSION O(1),R(1),C(1)
      INTEGER O,B,C
      C- THIS SUBROUTINE RETURNS THE LOCATION (I,J,K) OF THE FIRST K-VALUE ELEMENT
      C- ENCOUNTERED IN THE IMAGE. I-ROW ADDRESS(RECORD),J-WORD ADDRESS,II-ELEMENT
      C- ADDRESS.
      INTV(I1,J,II)=AND(SHIFT(O(I1)+J),-C(II),3)
      M=NWP-1
      I=1
10     BUFFER IN(IY,1) (O(1),O(NWP))
      IF (UNIT(IY).EQ.0) GO TO 40
      DO 10 J=1,M
      DO 20 II=2,31
      KK=INTV(I1,J,II)
      IF (KK.EQ.0) GO TO 30
      CONTINUE
      CONTINUE
      I=I+1
      GO TO 10
      I=J+1=0
      REMIND IY
      M=M-2
      DO 50 I=1,M
      BUFFER IN(IY,1) (O(1),O(NWP))
      IF (UNIT(IY).EQ.0) GO TO 30
      BUFFER OUT(IY,1) (O(1),O(NWP))
      L=UNIT(IY)
      BUFFER IN(IY,1) (O(1),O(NWP))
      IF (UNIT(IY).EQ.0) GO TO 30
      BUFFER IN(IY,1) (O(2),O(NWP-1))
      IF (UNIT(IY).EQ.0) GO TO 30
      BUFFER IN(IY,1) (O(3),O(NWP-1))
      IF (UNIT(IY).EQ.0) GO TO 30
      IF (UNIT(IY).EQ.0) GO TO 30
      I=I+1
      I=2
      RETURN
      CONTINUE
      OPT=110
      RETURN
      FORMAT(1H,*,ERROR IN FIRST---EOF ENCOUNTERED*)
      END

```


05/01/77 09.34.51

FTN 4.34414

SUBROUTINE PARFIND 74/74 001=1

```

60      L=J(1)+NWP-1
        IF(KK.NE.K) GO TO 91
        IF(IFLAG.EQ.0) CALL STORE(0,0,C,STACK,I,J,II,IP04,I00)
        IFLAG=0
        I=I+1
        GO TO 95

```

```

65      C-TEST LOWER RIGHT ELEMENT
        C-
        90      CONTINUE
        J1=J
        IF(C(II).EQ.0) J1=J+1
        KK=INTV(I+1,J1,II+1)
        IF(KK.NE.K) GO TO 95
        IF(IFLAG.EQ.0) CALL STORE(0,0,C,STACK,I,J,II,IP04,I00)
        IFLAG=0
        I=I+1
        J=J1
        II=II+1
        GO TO 95

```

```

75      C-TEST RIGHT ELEMENT
        C-
        95      CONTINUE
        J1=J
        IF(C(II).EQ.0) J1=J+1
        KK=INTV(I,J1,II+1)
        IF(KK.NE.K) GO TO 90
        IF(IFLAG.EQ.0) CALL STORE(0,0,C,STACK,I,J,II,IP04,I00)
        IFLAG=0
        J=J1
        II=II+1
        GO TO 95

```

```

80      C-TEST LEFT ELEMENT
        C-
        90      CONTINUE
        J1=J
        IF(C(II).EQ.0) J1=J-1
        KK=INTV(I,J1,II-1)
        IF(KK.NE.K) GO TO 95
        IF(IFLAG.EQ.0) CALL STORE(0,0,C,STACK,I,J,II,IP04,I00)
        J=J1
        II=II-1
        GO TO 90

```

```

95      C-EXECUTE BACKSTACK SEARCH
        C-
        95      CONTINUE
        CALL BKSTACK(STACK.TOP,I,J,II,IFLAG,INK,0,0,C,IP04,I00),RETURNS(200)

```

```

100      C-UPDATE FILES-GET PROPER RECORDS IN CONF
        C-
        100      CONTINUE
        CALL UPDATE(I,J,II,0,IV,IV1,NWC,IP04,I00,0)
        FOR=4AT(1H, *HERE I IS RARY*)
        GO TO 30

```

```

105      C-
        C-
        110      C-

```

```

110      C-
        C-
        110      C-

```

```

110      C-
        C-
        110      C-

```

```

110      C-
        C-
        110      C-

```

```

110      C-
        C-
        110      C-

```

06/01/77 09.34.51

FTN 4.54414

SUBROUTINE PAPERNO 76774 DDY=1

```

115      CONTINUE
120      DO 210 I=1, TOP
130      PRINT 100, (STACK(I), I=1, 30)
140      RETURN
150      STOP AT (14, 4) (15, 10)
160      END

```



```

1  SUBROUTINE STORE(I,J,STACK,I,J,II,IROM,IOP)
   DIMENSION O(I),M(I),C(I),STACK(1000,3)
   IFF=50 O,I,C,STACK,IOP
C-
C-THIS SUBROUTINE STORES THE ADDRESS OF AN ELEMENT ON THE STACK AND ZEROES
C-THAT ELEMENT OUT IN THE IMAGE
C-
C-
C-STORE ADDRESS
C-
   STACK(IOP,1)=IROM+I-1
   STACK(IOP,2)=J
   STACK(IOP,3)=II
   DO 10 L=1, IOP
   CONTINUE
   IOP=IOP+1
10  FORMAT(1H1,*SUBROUTINE STORE, IOP=*,14)
110  FORMAT(1H ,3(14,5X))
C-ZERO OUT ELEMENT IN IMAGE
C-
   O(I)=M(I)=C(I)=J1=SHIFT(CO*PL(3),3(11))
   RETURN
   END

```

```

1  SUBROUTINE UPDATE(I,J,L,O,IY,IYI,NP,NPOMS,IRON,N)
   DIMENSION O(1),O(1)
   INTEGER N,N
   C- THIS SUBROUTINE MAINTAINS PROPER INFORMATION IN COPE,KEEPS POINTERS AT
   C- CORRECT RELATIVE AND ABSOLUTE VALUES, AND MAINTAINS PROPER FILE STRUCTURE
   1000  FORMAT(1H,*,UPDATE=---IT IS HERE*)
   IF(I.EQ.2) GO TO 70
   C-
   C- CLEAR CODE
   C-
   BUFFER OUT(IYI,1) O(1),O(NPOMS)
   L=UNIT(IYI)
   BUFFER OUT(IYI,1) O(2),O(N(2)-1)
   L=UNIT(IYI)
   BUFFER OUT(IYI,1) O(3),O(N(3)+NPOMS-1)
   L=UNIT(IYI)
   N=IRON+7
   DO 10 IL=N,NPOMS
   BUFFER IN(IY,1) O(1),O(NPOMS)
   IF(UNIT(IY).EQ.0) GO TO 15
   BUFFER OUT(IYI,1) O(1),O(NPOMS)
   10  L=UNIT(IYI)
   15  CONTINUE
   ENDFILE IYI
   REWIND IY
   REWIND IYI
   C-
   C- CHANGE FILS
   C-
   L=IY
   IY=IYI
   IYI=L
   C-
   C- DETERMINE PROPER RECORDS TO HAVE IN COPE:SKIP FORWARD
   C-
   IRON=IRON+2
   N=IRON-1
   1010  FORMAT(1H,*,UPDATE DATA *,6(1F,5X))
   DO 20 IL=1,M
   BUFFER IN(IY,1) O(1),O(NPOMS)
   IF(UNIT(IY).EQ.0) GO TO 30
   BUFFER OUT(IYI,1) O(1),O(NPOMS)
   20  L=UNIT(IYI)
   L=1
   BUFFER IN(IY,1) O(1),O(NPOMS)
   IF(UNIT(IY).EQ.0) GO TO 40
   L=2
   BUFFER IN(IY,1) O(2),O(N(2)-1)
   IF(UNIT(IY).EQ.0) GO TO 40
   L=3
   BUFFER IN(IY,1) O(3),O(N(3)+NPOMS-1)
   IF(UNIT(IY).EQ.0) GO TO 40
   L=4
   70  CONTINUE
   RETURN

```

06/01/77 09.74.51

FTN 4.54414

SUBROUTINE UPDA1 74/4 OPT=1

```

75 CONTINUE
   BACKSPACE IV1
   GO TO 76 UNP=1.4
76 BACKSPACE IV
   CALL MUM1(0(7(2)),1,0(9(1)),1,NM0)
   CALL MUM1(0(1),1,0(9(2)),1,NM0)
   BUFFER IN(1,1)10(1),0(NM0))
   IF UNP1(1).EQ.0 GO TO 80
   1=2
   IRON=IRON-1
   RETURN
80 PRINT 100
81 STOP
82 PRINT 110
83 STOP
84 FORMAT(1H *,ERROR IN UPDATE WHILE INPUT TO CORE*)
85 FORMAT(1H *,ERROR IN UPDATE WHILE SKIPPING*)
86 END

```

06/01/77 09.14.51

ETN 4.5+414

SUBROUTINE MMH1 74/74 OPT=1

1 SUBROUTINE MMH1(A,IPROM,2,110,K)
C- THIS SUBROUTINE TRANSFERS K WORDS FROM CORE, BEGINNING WITH LOCATION A(IPROM).
C- TO CORE, BEGINNING WITH LOCATION 2(110).
C-

5 DIMENSION A(11),R(1)
INTEGER A,R
I=IPROM
J=110
10 DO 10 L=1,K
R(J)=A(I)
I=I+1
J=J+1
RETURN
END


```

1  SUBROUTINE STKST(STACK, TOP)
   DIMENSION STACK(1000,3)
   INTEGER STACK, TOP

5  C-THIS SUBROUTINE USES SUBROUTINES STP AND PRESRT TO SORT PIXEL STACK
   C-
   PRESRT=1
   N=1
   N=1
   IP=1
   IP=TOP
   CALL STP(STACK, IP, TOP, M)
   IF (IP.EQ.TOP.AND.M.GT.1) GO TO 10
   IF (M.NE.1) IR=IP+1
   CALL PRESRT(STACK, TOP, IR, TOP, M)
   N=N+1
   GO TO 10
10  CONTINUE
   IP=1
   N=N+1
   IF (N.LE.2) GO TO 20
   RETURN
END

```

SUBROUTINE PRESRT 74774 OPT=1

```

1  SUBROUTINE PRESRT(STACK, TOP, IP, TP, N)
   DIMENSION STACK(100,2)
   INTEGER STACK, TOP

C-
C- SUBROUTINE PRESRT FINDS THE EQUAL-VALUE ELEMENTS OF THE N TH COLUMN
C- OF STACK, AND DETERMINES THE LIMITS FOR SORT(IR, IP)
C-
   L=STACK(IP,N)
   IP=IP+1
10  IF (STACK(IP,N).GT.L) GO TO 20
   IF (N.GT.1.AND.STACK(IP+1).GT.STACK(IP,1)) GO TO 21
   IP=IP+1
   IF (IP.GT.TOP) GO TO 20
   GO TO 10
20  CONTINUE
   IP=IP-1
   RETURN
   END

```

74774. SUBCUTIVE SETS. $\alpha p r = 1$

REPORT THE SPEIS STACK, 12.1.11
INVESTION STACK 100.33

WIFE'S STOCK

IMPERIAL STOCK

Q-TWIS SORTS THE STACK OF PIXELS SORTING ON THE NTH COLUMN OF STACK BEGINNING WITH THE 1ST ELEMENT AND ENDING ON THE 1PTH ELEMENT

OF STACK, BEGINNING WITH THE 10TH ELEMENT AND ENDING ON THE 10TH ELEMENT

۷

NOTES ON THE

1991

05 20 Y=72.44

1 = T + 1

10 11 12 13 14 15 16 17 18 19 20 21 22 23 24 25 26 27 28 29 30 31 32 33 34 35 36 37 38 39 40 41 42 43 44 45 46 47 48 49 50 51 52 53 54 55 56 57 58 59 60 61 62 63 64 65 66 67 68 69 70 71 72 73 74 75 76 77 78 79 80 81 82 83 84 85 86 87 88 89 90 91 92 93 94 95 96 97 98 99 100 101 102 103 104 105 106 107 108 109 110 111 112 113 114 115 116 117 118 119 120 121 122 123 124 125 126 127 128 129 130 131 132 133 134 135 136 137 138 139 140 141 142 143 144 145 146 147 148 149 150 151 152 153 154 155 156 157 158 159 160 161 162 163 164 165 166 167 168 169 170 171 172 173 174 175 176 177 178 179 180 181 182 183 184 185 186 187 188 189 190 191 192 193 194 195 196 197 198 199 200 201 202 203 204 205 206 207 208 209 210 211 212 213 214 215 216 217 218 219 220 221 222 223 224 225 226 227 228 229 230 231 232 233 234 235 236 237 238 239 240 241 242 243 244 245 246 247 248 249 250 251 252 253 254 255 256 257 258 259 260 261 262 263 264 265 266 267 268 269 270 271 272 273 274 275 276 277 278 279 280 281 282 283 284 285 286 287 288 289 290 291 292 293 294 295 296 297 298 299 300 301 302 303 304 305 306 307 308 309 310 311 312 313 314 315 316 317 318 319 320 321 322 323 324 325 326 327 328 329 330 331 332 333 334 335 336 337 338 339 340 341 342 343 344 345 346 347 348 349 350 351 352 353 354 355 356 357 358 359 360 361 362 363 364 365 366 367 368 369 370 371 372 373 374 375 376 377 378 379 380 381 382 383 384 385 386 387 388 389 390 391 392 393 394 395 396 397 398 399 400 401 402 403 404 405 406 407 408 409 410 411 412 413 414 415 416 417 418 419 420 421 422 423 424 425 426 427 428 429 430 431 432 433 434 435 436 437 438 439 440 441 442 443 444 445 446 447 448 449 450 451 452 453 454 455 456 457 458 459 460 461 462 463 464 465 466 467 468 469 470 471 472 473 474 475 476 477 478 479 480 481 482 483 484 485 486 487 488 489 490 491 492 493 494 495 496 497 498 499 500 501 502 503 504 505 506 507 508 509 510 511 512 513 514 515 516 517 518 519 520 521 522 523 524 525 526 527 528 529 530 531 532 533 534 535 536 537 538 539 540 541 542 543 544 545 546 547 548 549 550 551 552 553 554 555 556 557 558 559 560 561 562 563 564 565 566 567 568 569 570 571 572 573 574 575 576 577 578 579 580 581 582 583 584 585 586 587 588 589 590 591 592 593 594 595 596 597 598 599 600 601 602 603 604 605 606 607 608 609 610 611 612 613 614 615 616 617 618 619 620 621 622 623 624 625 626 627 628 629 630 631 632 633 634 635 636 637 638 639 640 641 642 643 644 645 646 647 648 649 650 651 652 653 654 655 656 657 658 659 660 661 662 663 664 665 666 667 668 669 670 671 672 673 674 675 676 677 678 679 680 681 682 683 684 685 686 687 688 689 690 691 692 693 694 695 696 697 698 699 700 701 702 703 704 705 706 707 708 709 710 711 712 713 714 715 716 717 718 719 720 721 722 723 724 725 726 727 728 729 730 731 732 733 734 735 736 737 738 739 740 741 742 743 744 745 746 747 748 749 750 751 752 753 754 755 756 757 758 759 760 761 762 763 764 765 766 767 768 769 770 771 772 773 774 775 776 777 778 779 780 781 782 783 784 785 786 787 788 789 790 791 792 793 794 795 796 797 798 799 800 801 802 803 804 805 806 807 808 809 810 811 812 813 814 815 816 817 818 819 820 821 822 823 824 825 826 827 828 829 830 831 832 833 834 835 836 837 838 839 840 841 842 843 844 845 846 847 848 849 850 851 852 853 854 855 856 857 858 859 860 861 862 863 864 865 866 867 868 869 870 871 872 873 874 875 876 877 878 879 880 881 882 883 884 885 886 887 888 889 890 891 892 893 894 895 896 897 898 899 900 901 902 903 904 905 906 907 908 909 910 911 912 913 914 915 916 917 918 919 920 921 922 923 924 925 926 927 928 929 930 931 932 933 934 935 936 937 938 939 940 941 942 943 944 945 946 947 948 949 950 951 952 953 954 955 956 957 958 959 960 961 962 963 964 965 966 967 968 969 970 971 972 973 974 975 976 977 978 979 980 981 982 983 984 985 986 987 988 989 990 991 992 993 994 995 996 997 998 999 1000 1001 1002 1003 1004 1005 1006 1007 1008 1009 1010 1011 1012 1013 1014 1015 1016 1017 1018 1019 1020 1021 1022 1023 1024 1025 1026 1027 1028 1029 1030 1031 1032 1033 1034 1035 1036 1037 1038 1039 1040 1041 1042 1043 1044

RECYCCLING, LEAD STACK(J,N) GO TO 10

15 12 11 10 9 8 7 6 5 4 3 2 1

43-15741-10

```

404 = STACK(I, I)
STACK(I, I) = STACK(I, I)

```

[illegible]

CONTINUED

CONTINUED

CONFIDENTIAL

July 2

```

1  SUBROUTINE INTNOTN(STACK, TOP, L, LL)
2  DIMENSION STACK(1000, 3)
3  INTEGER STACK, TOP, R, EC, IP
4
5  C- THIS SUBROUTINE PUTS THE ELEMENTS OF THE STACK INTO INTERVAL NOTATION
6
7  C-
8
9  LL=L
10 IP=1
11 CALL PRESPT(STACK, TOP, IR, IP, 1)
12 R=STACK(IR, 1)
13 EC=10=STACK(IR, 2)+3)*STACK(IR, 3)
14 IP=IP+1
15 IF (STACK(IR, 2).GT. STACK(IR, 3)) GO TO 30
16 IF (STACK(IR, 3)-STACK(IR, 1).NE. 1) GO TO 40
17 R=STACK(IR, 3)-STACK(IR, 1, 3)
18 EC=EC+1
19 I=I+1
20 IF (I.GT. TOP) GO TO 40
21 IF (I.GT. TOP) GO TO 40
22 GO TO 20
23 IF (STACK(I, 3).NE. 2.0)*STACK(I-1, 3).NE. 1) GO TO 40
24 GO TO 25
25 CONTINUE
26 IF (I.GT. I) PRINT*, " UNDERFLOW IN INTNOTN ", L, I
27 STACK(I, 1)=R
28 STACK(I, 2)=EC
29 STACK(I, 3)=EC
30 IF (I.GT. TOP) GO TO 51
31 IF (I.GT. IP) GO TO 50
32 EC=EC+STACK(I, 2)+3)*STACK(I, 3)
33 L=L+1
34 LL=LL+1
35 IR=I
36 I=I+1
37 GO TO 20
38 CONTINUE
39 IP=IP+1
40 L=L+1
41 LL=LL+1
42 GO TO 10
43 CONTINUE
44 RETURN
45 END

```



```

1  SUBROUTINE SURM02S(O,C,STACK,TOP,L,NEWS,NWR,IV,IY1,IR0W)
   DIMENSION O(1),C(1),I(1),STACK(1000,3)
   ENTERED O,O,C,STACK,TOP,POINT2
   IMPRINT OUTIN1
5  IF (I.II) AND (SHIFT(I,IR0W)+J).C(I,I),3)
   C- ASSUMED LAST INTERVAL OF PARTICLE TO BE IN 2ND RECORD LOCATION IN CORE (D)
   C-
   C- GENERATE WORKING STACK: POINT1 IS POINTER TO TOP OF WORKING STACK: POINT2 IS
   C- POINTER TO TOP OF STORAGE STACK.
   C-
   PRINT *, "SURM02S"
   PRINT *, "L = ", L
   TOP=L
   I=STACK(L,1)-I*OW+1
   CALL LOCATE(I,J,I,I,O,IV,IY1,NWR,NEWS,I*OW,R)
   POINT1=L
   I=STACK(L,2)
   N=STACK(L,3)
   DO 10 I1=N,N
   POINT1=POINT1+1
   STACK(POINT1,1)=I*OW+1
   STACK(POINT1,2)=I1/10
   STACK(POINT1,3)=MOD(I1,10)+1
10  CONTINUE
   POINT2=POINT1+1
   C-
   C- EXECUTE LOWER RECORD SEARCH: PLACING ELEMENTS ON STORAGE STACK
   C-
   I=7
   M=L+1
   L=POINT1
   DO 30 I1=4,POINT1
15  C- SET POINTERS TO WORD AND SHIFT ARRAYS
   C-
   J=STACK(I1,2)
   II=STACK(I1,3)
   C- TEST LOWER LEFT ELEMENT
   C-
   J1=J
   IF (C(I1),EQ,50) J1=J-1
   KK=INTV(I,J1,II-1)
   IF (KK,NE,2) GO TO 30
   J=J1
   II=II-1
   CALL STORE(O,O,C,STACK,I,J,II,I*OW,POINT2)
   GO TO 50
30  CONTINUE
   C-
   C- TEST LOWER ELEMENT
   C-
   KK=INTV(I,J,II)
   IF (KK,NE,2) GO TO 40
   CALL STORE(O,O,C,STACK,I,J,II,I*OW,POINT2)
   GO TO 50
40  CONTINUE
45  C-

```

```

C-TEST LOWER RIGHT ELEMENT
C-
60      JI=J
        IF(C(IJ).EQ.0) JI=J+1
        KK=INTV(I,J,I,II+1)
        IF(KK.NE.2) GO TO 69
        J=JI
        II=II+1
        CALL STORE(D,B,C,STACK,I,J,II,IPROW,POINT2)
        CONTINUE
50      C-
C-SEARCH LEFT
C-
70      JI=J
        IF(C(IJ).EQ.5A) JI=J-1
        KK=INTV(II,J,I,II-1)
        IF(KK.NE.2) GO TO 69
        C-ABOVE MIGHT HAVE TO BE CONNECTED IN I,J,II
        J=JI
        II=II-1
        CALL STORE(D,B,C,STACK,I,J,II,IPROW,POINT2)
        GO TO 59
60      CONTINUE
C-
C-SEARCH RIGHT
C-
80      JI=J
        IF(C(IJ).EQ.0) JI=J+1
        KK=INTV(II,J,I,II+1)
        IF(KK.NE.2) GO TO 99
        J=JI
        II=II+1
        CALL STORE(D,B,C,STACK,I,J,II,IPROW,POINT2)
        GO TO 60
90      CONTINUE
C-
C-TEST FOR EMPTY STORAGE STACK
C-
95      IF(POINT2.EQ.POINT1+1) GO TO 99
        C-SHIFT WORKING STACK:REPEAT SEARCH
        M=POINT1+1
        POINT1=POINT2-1
100     C-
C-SHIFT RECORDS IN CORE:INPUT NEXT RECORD
C-
        BUFFER OUT(IY1,1)(C(I),J(NH0))
        NC=UNIT(IY1)
        CALL MWM1(J(P(2)),1,0(1),1,NH0)
        CALL MWM1(0(8(3)),1,0(8(2)),1,NH0)
        BUFFER IN(IY,1)(C(I(3)),C(I(2))+NH0-1)
        IF(UNIT(IY)) 85,97,95
        IPROW=IP04+1
        GO TO 15
95      CONTINUE
97      POINT="RECORDS IN SUBROUTINE"
        STOP
99      CONTINUE

```

06/11/77 09.14.51

RTN 4.5+014

SUBROUTINE SUBROUTINE 7/74 OPT=1

```

115      M=MIN(L+1
      M=MIN(M,L)
      CALL STAGE(STACK(L+1),M)
      M=TOP-1
      CALL INTOUT(STACK(L+1),M,M,LL)
      M=TOP+1
      TOP=L
      RETURN
      END

```

```

1  SUBROUTINE TEST(D,R,C,STACK,TOP,L,NROWS,NW2,IV,IY1,IPOW)
   DIMENSION D(1),R(1),C(1),STACK(1000,3)
   INTEGER D,R,C,STACK,TOP,POINT2
   INTEGER POINT1
   INTV(I,J,II)=ANNUSHTFT((R(II)+J)-C(II),3)
   C- ASSUMED LAST INTERVAL OF PARTICLE TO BE IN 2ND DEGREE LOCATION IN CORE (R)
   C-
   C- GENERATE WORKING STACK: POINT1 IS POINTED TO TOP OF WORKING STACK: POINT2 IS
   C- POINTED TO TOP OF STORAGE STACK
   C-
   TOP=L
   I=STACK(I,1)-IPOW+1
   CALL UPDATE(I,J,II,D,IV,IY1,NW2,NROWS,IPOW,R)
   POINT1=L
   M=STACK(I,2)
   N=STACK(I,3)
   DO 10 II=4,N
     POINT1=POINT1 + 1
     STACK(POINT1,1)=IPOW
     STACK(POINT1,2)=I/30
     STACK(POINT1,3)=M+(I,30)*1
   CONTINUE
   POINT2=POINT1+1
10  POINT2=POINT1+1
   C-
   C- EXECUTE LOWER RECOGN SEARCH: PLACING ELEMENTS ON STORAGE STACK
   C-
   I=1
   M=L+1
   L=POINT1
   DO 10 II=1,POINT1
15  C-
   C- SET POINTERS TO WORD AND SHIFT APOAYS
   C-
   J=STACK(II,2)
   II=STACK(II,3)
   C-
   C- TEST LOWER LEFT ELEMENT
   C-
   J1=J
   IF(C(II).EQ.59) J1=J-1
   KK=INTV(I,J1,II-1)
   IF(KK.NE.2) GO TO 10
   J=J1
   II=II-1
   CALL STOPS(D,R,C,STACK,I,J,II,IPOW,POINT2)
   GO TO 50
   CONTINUE
30  C-
   C- TEST LOWER ELEMENT
   C-
   KK=INTV(I,J,II)
   IF(KK.NE.2) GO TO 40
   CALL STOPS(D,R,C,STACK,I,J,II,IPOW,POINT2)
   GO TO 50
   CONTINUE
40  C-
   C- TEST LOWER RIGHT ELEMENT
   C-

```


PAGE 3

06/01/77 00.34.51

FTI 4.54416

74/74 001=1

SUBROUTINE TEST

115

END

```

C-PROGRAM TYPED OUT(INPUT CONTINUT,IACF(I)
      DIMENSION IPI(20),LTY(11),Z,ICHAR(64)
      NAMELIST/INPDS,NCOLS,IROW,ICOL
C-THIS PROGRAM PRINTS THE THRESHOLD IMAGE OF 8-BIT PIXELS ON THE LINE
C-DURING ASSIGNING EACH INTENSITY VALUE A CHARACTER.DETERMINED FROM INPUT CARD.
C-INPUT PARAMETERS
C-INPDS---LAST RECORD OF IMAGE TO BE PRINTED(ASSUMES 500)
C-NCOLS---LAST SAMPLE(PIXEL) OF IMAGE TO BE PRINTED (ASSUMES 500)
C-IROW---STARTING RECORD OF IMAGE TO BE PRINTED (ASSUMES 1)
C-ICOL---STARTING CAMPLE(PIXEL) OF EACH RECORD TO BE PRINTED(ASSUMES 1)
C-ICMAP---ARRAY OF 64 CHARACTERS TO REPRESENT INTENSITIES 0-63 RESPECTIVELY ON
      PRINTED IMAGE
C-ASSIGN DEFAULT VALUES TO *SIZE* PARAMETERS
      NROWS=NCOLS=500
      LTY=ICOL=1
C-CARD INPUT PHASE:PRINT INPUT PARAMETERS
      PRINT 190
      READ SIZE
      PRINT SIZE
      READ 110,ICMAP(1),I=1,63
      PRINT 120
      DO 10 I=1,9
        PRINT 130,I-1,ICMAP(I),I+7,ICMAP(I+9),I+15,ICMAP(I+15),I+23,
          ICHAR(I+29),I+31,ICMAP(I+37),I+39,ICMAP(I+40),I+47,ICMAP(I+49).
        ZI+55,ICMAP(I+56)
      CONTINUE
10    CONTINUE
C-DETERMINING RECORD LENGTH IN WORDS AND START PAGE LOOP
      NWR=NCOLS/10
      IF(INWR*10.LT.NCOLS)NWF=NWR+1
      PRINT 140
      N=10
      DO 50 IT=ICOL,NCOLS,170
C-SKIP FORWARD TO FIRST RECORD
C-
      IF(IROW.EQ.1) GO TO 20
      L=IROW-1
      DO 15 I=L
        GUESS IN(1,1)(IN(1),TN(2))
        IF(CUNIT(1))15,999,15
      CONTINUE
      CONTINUE
      PRINT 100
      DO 55 I=IROW,NROWS
        GUESS IN(1,1)(IN(1),I*(NWR))
        IF(CUNIT(1))30,999,60
      CONTINUE
      IF((I+6).GT.NCOLS)H=NCOLS-I+1

```

```

60      CALL UNPACK(IN,LTNF,IT,I)
        DO 40 J=1,N
          LINF(J)=ICCHAR(LINF(J),I)
          PRINT 150,(LINF(J),J=1,4)
          CONTINUE
          REMAIN 1
          CONTINUE
          STOP
          CONTINUE
          PRINT *, " EOF ON TAPE1, RECORD = ", I
          STOP
          CONTINUE
          100 FORMAT(IH1)
          110 FORMAT(64A1)
          120 FORMAT(11H1,"TABLE OF CHARACTERS USED FOR INTENSITIES  *////")
          130 FORMAT(11H0,I2," - ",A1,7(I9X,I2," - ",A1))
          140 FORMAT(11H1)
          150 FORMAT(11H ,130A1)
          END
75

```



```

1      PROGRAM PARSED(INPUT,OUTPUT,TAPE3,TAPE4,TAPE5)
2
3      C-PROGRAM PARSED GENERATES THE INTERIOR PARTICLE VALUES IN THE THRESHOLDED IMAGE
4
5      C-
6      C-
7      C-NROWS-- THE NUMBER OF ROWS IN INPUT IMAGE
8      C-NCOLS-- THE NUMBER OF COLUMNS PER ROW IN THE INPUT IMAGE
9      C-TAPE3-- CONTAINS THE THRESHOLDED IMAGE (INPUT).
10     C-TAPE5-- USED AS SCRATCH FILE
11     C-TAPE4-- CONTAINS THE OUTPUT IMAGE: WITH INTERIOR VALUES GENERATED.
12
13     C-
14     DIMENSION B(7),C(32),O(1300)
15     INTEGER B,C,O
16     READ INPUT,NROWS,NCOLS
17     FORMAT(15)
18     IFLAG=0
19     IY=6
20     IY1=6
21     NM2=NCOLS/2
22
23     C-INITIALIZE I POINTER ARRAY (3) AND SHIFT ARRAY (C)
24     C-
25
26     I(1)=1
27     B(2)=NWP+1
28     B(3)=B(2)+NMR
29     C(1)=58
30     C(1)=0
31     KE=50
32     DO 10 I=2,31
33     KE=KE-2
34     C(I)=KE
35     K=1
36     PRINT 5000,C
37     PRINT 5000,B
38     FORMAT(1H1,11I4)
39
40     INITIALIZE FIRST "RECORD" OF O TO 0'S
41     O(1)=0
42     CALL MMV1(O,1,O,2,NWP-1)
43
44     CONTINUE
45
46     C-START FIRST PASS--FIND AND CONVERT PROPER 1-VALUED PIXELS TO 2-VALUED PIXELS
47
48     C-
49     C-
50     C-
51     C-
52     C-
53     C-
54     C-
55     C-
56     C-
57     C-
58     C-
59     C-
60     C-
61     C-
62     C-
63     C-
64     C-
65     C-
66     C-
67     C-
68     C-
69     C-
70     C-
71     C-
72     C-
73     C-
74     C-
75     C-
76     C-
77     C-
78     C-
79     C-
80     C-
81     C-
82     C-
83     C-
84     C-
85     C-
86     C-
87     C-
88     C-
89     C-
90     C-
91     C-
92     C-
93     C-
94     C-
95     C-
96     C-
97     C-
98     C-
99     C-
100    C-
101    C-
102    C-
103    C-
104    C-
105    C-
106    C-
107    C-
108    C-
109    C-
110    C-
111    C-
112    C-
113    C-
114    C-
115    C-
116    C-
117    C-
118    C-
119    C-
120    C-
121    C-
122    C-
123    C-
124    C-
125    C-
126    C-
127    C-
128    C-
129    C-
130    C-
131    C-
132    C-
133    C-
134    C-
135    C-
136    C-
137    C-
138    C-
139    C-
140    C-
141    C-
142    C-
143    C-
144    C-
145    C-
146    C-
147    C-
148    C-
149    C-
150    C-
151    C-
152    C-
153    C-
154    C-
155    C-
156    C-
157    C-
158    C-
159    C-
160    C-
161    C-
162    C-
163    C-
164    C-
165    C-
166    C-
167    C-
168    C-
169    C-
170    C-
171    C-
172    C-
173    C-
174    C-
175    C-
176    C-
177    C-
178    C-
179    C-
180    C-
181    C-
182    C-
183    C-
184    C-
185    C-
186    C-
187    C-
188    C-
189    C-
190    C-
191    C-
192    C-
193    C-
194    C-
195    C-
196    C-
197    C-
198    C-
199    C-
200    C-
201    C-
202    C-
203    C-
204    C-
205    C-
206    C-
207    C-
208    C-
209    C-
210    C-
211    C-
212    C-
213    C-
214    C-
215    C-
216    C-
217    C-
218    C-
219    C-
220    C-
221    C-
222    C-
223    C-
224    C-
225    C-
226    C-
227    C-
228    C-
229    C-
230    C-
231    C-
232    C-
233    C-
234    C-
235    C-
236    C-
237    C-
238    C-
239    C-
240    C-
241    C-
242    C-
243    C-
244    C-
245    C-
246    C-
247    C-
248    C-
249    C-
250    C-
251    C-
252    C-
253    C-
254    C-
255    C-
256    C-
257    C-
258    C-
259    C-
260    C-
261    C-
262    C-
263    C-
264    C-
265    C-
266    C-
267    C-
268    C-
269    C-
270    C-
271    C-
272    C-
273    C-
274    C-
275    C-
276    C-
277    C-
278    C-
279    C-
280    C-
281    C-
282    C-
283    C-
284    C-
285    C-
286    C-
287    C-
288    C-
289    C-
290    C-
291    C-
292    C-
293    C-
294    C-
295    C-
296    C-
297    C-
298    C-
299    C-
300    C-
301    C-
302    C-
303    C-
304    C-
305    C-
306    C-
307    C-
308    C-
309    C-
310    C-
311    C-
312    C-
313    C-
314    C-
315    C-
316    C-
317    C-
318    C-
319    C-
320    C-
321    C-
322    C-
323    C-
324    C-
325    C-
326    C-
327    C-
328    C-
329    C-
330    C-
331    C-
332    C-
333    C-
334    C-
335    C-
336    C-
337    C-
338    C-
339    C-
340    C-
341    C-
342    C-
343    C-
344    C-
345    C-
346    C-
347    C-
348    C-
349    C-
350    C-
351    C-
352    C-
353    C-
354    C-
355    C-
356    C-
357    C-
358    C-
359    C-
360    C-
361    C-
362    C-
363    C-
364    C-
365    C-
366    C-
367    C-
368    C-
369    C-
370    C-
371    C-
372    C-
373    C-
374    C-
375    C-
376    C-
377    C-
378    C-
379    C-
380    C-
381    C-
382    C-
383    C-
384    C-
385    C-
386    C-
387    C-
388    C-
389    C-
390    C-
391    C-
392    C-
393    C-
394    C-
395    C-
396    C-
397    C-
398    C-
399    C-
400    C-
401    C-
402    C-
403    C-
404    C-
405    C-
406    C-
407    C-
408    C-
409    C-
410    C-
411    C-
412    C-
413    C-
414    C-
415    C-
416    C-
417    C-
418    C-
419    C-
420    C-
421    C-
422    C-
423    C-
424    C-
425    C-
426    C-
427    C-
428    C-
429    C-
430    C-
431    C-
432    C-
433    C-
434    C-
435    C-
436    C-
437    C-
438    C-
439    C-
440    C-
441    C-
442    C-
443    C-
444    C-
445    C-
446    C-
447    C-
448    C-
449    C-
450    C-
451    C-
452    C-
453    C-
454    C-
455    C-
456    C-
457    C-
458    C-
459    C-
460    C-
461    C-
462    C-
463    C-
464    C-
465    C-
466    C-
467    C-
468    C-
469    C-
470    C-
471    C-
472    C-
473    C-
474    C-
475    C-
476    C-
477    C-
478    C-
479    C-
480    C-
481    C-
482    C-
483    C-
484    C-
485    C-
486    C-
487    C-
488    C-
489    C-
490    C-
491    C-
492    C-
493    C-
494    C-
495    C-
496    C-
497    C-
498    C-
499    C-
500    C-
501    C-
502    C-
503    C-
504    C-
505    C-
506    C-
507    C-
508    C-
509    C-
510    C-
511    C-
512    C-
513    C-
514    C-
515    C-
516    C-
517    C-
518    C-
519    C-
520    C-
521    C-
522    C-
523    C-
524    C-
525    C-
526    C-
527    C-
528    C-
529    C-
530    C-
531    C-
532    C-
533    C-
534    C-
535    C-
536    C-
537    C-
538    C-
539    C-
540    C-
541    C-
542    C-
543    C-
544    C-
545    C-
546    C-
547    C-
548    C-
549    C-
550    C-
551    C-
552    C-
553    C-
554    C-
555    C-
556    C-
557    C-
558    C-
559    C-
560    C-
561    C-
562    C-
563    C-
564    C-
565    C-
566    C-
567    C-
568    C-
569    C-
570    C-
571    C-
572    C-
573    C-
574    C-
575    C-
576    C-
577    C-
578    C-
579    C-
580    C-
581    C-
582    C-
583    C-
584    C-
585    C-
586    C-
587    C-
588    C-
589    C-
590    C-
591    C-
592    C-
593    C-
594    C-
595    C-
596    C-
597    C-
598    C-
599    C-
600    C-
601    C-
602    C-
603    C-
604    C-
605    C-
606    C-
607    C-
608    C-
609    C-
610    C-
611    C-
612    C-
613    C-
614    C-
615    C-
616    C-
617    C-
618    C-
619    C-
620    C-
621    C-
622    C-
623    C-
624    C-
625    C-
626    C-
627    C-
628    C-
629    C-
630    C-
631    C-
632    C-
633    C-
634    C-
635    C-
636    C-
637    C-
638    C-
639    C-
640    C-
641    C-
642    C-
643    C-
644    C-
645    C-
646    C-
647    C-
648    C-
649    C-
650    C-
651    C-
652    C-
653    C-
654    C-
655    C-
656    C-
657    C-
658    C-
659    C-
660    C-
661    C-
662    C-
663    C-
664    C-
665    C-
666    C-
667    C-
668    C-
669    C-
670    C-
671    C-
672    C-
673    C-
674    C-
675    C-
676    C-
677    C-
678    C-
679    C-
680    C-
681    C-
682    C-
683    C-
684    C-
685    C-
686    C-
687    C-
688    C-
689    C-
690    C-
691    C-
692    C-
693    C-
694    C-
695    C-
696    C-
697    C-
698    C-
699    C-
700    C-
701    C-
702    C-
703    C-
704    C-
705    C-
706    C-
707    C-
708    C-
709    C-
710    C-
711    C-
712    C-
713    C-
714    C-
715    C-
716    C-
717    C-
718    C-
719    C-
720    C-
721    C-
722    C-
723    C-
724    C-
725    C-
726    C-
727    C-
728    C-
729    C-
730    C-
731    C-
732    C-
733    C-
734    C-
735    C-
736    C-
737    C-
738    C-
739    C-
740    C-
741    C-
742    C-
743    C-
744    C-
745    C-
746    C-
747    C-
748    C-
749    C-
750    C-
751    C-
752    C-
753    C-
754    C-
755    C-
756    C-
757    C-
758    C-
759    C-
760    C-
761    C-
762    C-
763    C-
764    C-
765    C-
766    C-
767    C-
768    C-
769    C-
770    C-
771    C-
772    C-
773    C-
774    C-
775    C-
776    C-
777    C-
778    C-
779    C-
780    C-
781    C-
782    C-
783    C-
784    C-
785    C-
786    C-
787    C-
788    C-
789    C-
790    C-
791    C-
792    C-
793    C-
794    C-
795    C-
796    C-
797    C-
798    C-
799    C-
800    C-
801    C-
802    C-
803    C-
804    C-
805    C-
806    C-
807    C-
808    C-
809    C-
810    C-
811    C-
812    C-
813    C-
814    C-
815    C-
816    C-
817    C-
818    C-
819    C-
820    C-
821    C-
822    C-
823    C-
824    C-
825    C-
826    C-
827    C-
828    C-
829    C-
830    C-
831    C-
832    C-
833    C-
834    C-
835    C-
836    C-
837    C-
838    C-
839    C-
840    C-
841    C-
842    C-
843    C-
844    C-
845    C-
846    C-
847    C-
848    C-
849    C-
850    C-
851    C-
852    C-
853    C-
854    C-
855    C-
856    C-
857    C-
858    C-
859    C-
860    C-
861    C-
862    C-
863    C-
864    C-
865    C-
866    C-
867    C-
868    C-
869    C-
870    C-
871    C-
872    C-
873    C-
874    C-
875    C-
876    C-
877    C-
878    C-
879    C-
880    C-
881    C-
882    C-
883    C-
884    C-
885    C-
886    C-
887    C-
888    C-
889    C-
890    C-
891    C-
892    C-
893    C-
894    C-
895    C-
896    C-
897    C-
898    C-
899    C-
900    C-
901    C-
902    C-
903    C-
904    C-
905    C-
906    C-
907    C-
908    C-
909    C-
910    C-
911    C-
912    C-
913    C-
914    C-
915    C-
916    C-
917    C-
918    C-
919    C-
920    C-
921    C-
922    C-
923    C-
924    C-
925    C-
926    C-
927    C-
928    C-
929    C-
930    C-
931    C-
932    C-
933    C-
934    C-
935    C-
936    C-
937    C-
938    C-
939    C-
940    C-
941    C-
942    C-
943    C-
944    C-
945    C-
946    C-
947    C-
948    C-
949    C-
950    C-
951    C-
952    C-
953    C-
954    C-
955    C-
956    C-
957    C-
958    C-
959    C-
960    C-
961    C-
962    C-
963    C-
964    C-
965    C-
966    C-
967    C-
968    C-
969    C-
970    C-
971    C-
972    C-
973    C-
974    C-
975    C-
976    C-
977    C-
978    C-
979    C-
980    C-
981    C-
982    C-
983    C-
984    C-
985    C-
986    C-
987    C-
988    C-
989    C-
990    C-
991    C-
992    C-
993    C-
994    C-
995    C-
996    C-
997    C-
998    C-
999    C-
1000   C-

```

06/01/77 09.31.42

FTN 4.54414

PROGRAM PAUSED 74/74 OPT=1

```

C-USE INT123 TO SEARCH FOR 2'S AND 3'S (CONVERSION)
C-
25 CONTINUE
   CALL INT123(I0,R,C,K,NWP)
C-
C-WRITE OUT FIRST "RECORD" FROM 0
   BUFFER OUT(IY1,1) (0(1),0(NWP))
   IWAIT=UNIT(IY1)
   IF(FLAG.EQ.1) GO TO 30
C-
C-SHIFT 0 TACK UP 1 "RECORD", AND BUFFER IN NEXT RECORD
C-
   CALL MMW1(I0,R(2),0,1,NWP)
   CALL MMW1(I0,R(3),0,2,NWP)
   BUFFER IN (IY,1) (0(R(3)+1),0(R(3)+NWP-1))
   IF(UNIT(IY).NE.0) GO TO 20
   FLAG=1
   0(R(3))=0
   CALL MMW1(I0,R(3),0,3(3)+1,NWP-1)
   GO TO 20
30  BUFFER OUT(IY1,1) (0(R(2)),0(R(3)-1))
   IWAIT=UNIT(IY1)
   BUFFER OUT(IY1,1) (0(R(3)),0(R(3)+NWP-1))
   IWAIT=UNIT(IY1)
   ENDFILE IY1
   REWIND IY
   REWIND IY1
C-
C-REPEAT ABOVE PROCEDURE FOR SECOND PASS: FIND AND CONVERT PROPER 2-VALUED PIXEL
C- TO 3-VALUED PIXELS
C-
   IY1=3
   IY=4
   K=2
   BUFFER IN(IY,1) (0(1),0(NWP))
   IWAIT=UNIT(IY)
   FLAG=0
   K=2
   BUFFER IN(IY,1) (0(R(2)),0(R(3)-1))
   IWAIT=UNIT(IY)
   BUFFER IN(IY,1) (0(R(3)),0(R(3)+NWP-1))
   IWAIT=UNIT(IY)
   CALL INT123(I0,R,C,K,NWP)
   BUFFER OUT(IY1,1) (0(1),0(NWP))
   IWAIT=UNIT(IY1)
   IF(FLAG.EQ.1) GO TO 50
   CALL MMW1(I0,R(2),0,1,NWP)
   CALL MMW1(I0,R(3),0,2(NWP))
   BUFFER IN(IY,1) (0(R(3)),0(R(3)+NWP-1))
   IF(UNIT(IY).NE.0) GO TO 40
   FLAG=1
   0(R(3))=0
   CALL MMW1(I0,R(3),0,3(3)+1,NWP-1)
   GO TO 40
50  CONTINUE
   BUFFER OUT(IY1,1) (0(R(2)),0(R(3)-1))
   IWAIT=UNIT(IY1)

```

PROGRAM PART 7.74 OPT=1

END OF TV1
REVIEW TV1
REVIEW TV1
REVIEW TV1
STOP
END

115

116

```

1  SUBROUTINE INIT2(A,B,C,N,M)
C- THIS SUBROUTINE GENERATES THE INTERIOR VALUES FOR THE PARTICLES. ON THE FIRST
C- THROUGH THE IMAGE. IT CONVERTS 1-VALUED PIXELS TO 2-VALUED PIXELS WHERE THE
C- BORDER CONDITIONS ARE PRESENT. ON THE SECOND PASS THROUGH THE IMAGE, 2-VALUED
C- PIXELS ARE CONVERTED TO 1-VALUED PIXELS WHERE THE BORDER CONDITIONS ARE
C- PRESENT.
C-
C- DIMENSION A(1),B(1),C(1)
C- INTEGER A,B,C
C- INTV(1,J,1)=AND(SHIFT(1,(B(1)+J),-C(1)),3)
C- MSK1=3
C- MSK2=COMPL(MSK1)
C- MM=4-1
C- I=2
C- K=N
C- DO 10 J=1,MM
C- DO 10 IT=2,31
C- KK=INTV(1,J,IT)
C- IF(KK,LT,K) GO TO 10
C- IF(K,EQ,1) GO TO 5
C-
C- C-TEST UPPER LEFT ELEMENT
C-
C- IF(C(1),NE,5) KK=INTV(1-1,J,IT-1)
C- IF(C(1),EQ,5) KK=INTV(1-1,J-1,IT-1)
C- IF(KK,LT,K) GO TO 10
C-
C- C-TEST UPPER RIGHT ELEMENT
C-
C- IF(C(1),NE,0) KK=INTV(1-1,J,IT+1)
C- IF(C(1),EQ,0) KK=INTV(1-1,J+1,IT+1)
C- IF(KK,LT,K) GO TO 10
C-
C- TEST LOWER LEFT ELEMENT
C-
C- IF(C(1),NE,5) KK=INTV(1+1,J,IT-1)
C- IF(C(1),EQ,5) KK=INTV(1+1,J-1,IT-1)
C- IF(KK,LT,K) GO TO 10
C-
C- C-TEST LOWER RIGHT ELEMENT
C-
C- IF(C(1),NE,0) KK=INTV(1+1,J,IT+1)
C- IF(C(1),EQ,0) KK=INTV(1+1,J+1,IT+1)
C- IF(KK,LT,K) GO TO 10
C-
C- C-TEST UPPER ELEMENT
C-
C- KK=INTV(1-1,J,IT)
C- IF(KK,LT,K) GO TO 10
C-
C- C-TEST RIGHT ELEMENT
C-
C- IF(C(1),NE,0) KK=INTV(1,J,IT+1)
C- IF(C(1),EQ,0) KK=INTV(1,J+1,IT+1)
C- IF(KK,LT,K) GO TO 10
C-

```


16/01/77 09.31.42

FTN 4.5+414

SUBROUTINE IHT2 DATA OPT=1

C-TEST LAST ELEMENT

```

C-      IF (I+1).NE.50)KK=INTV(I,J,I)-1
      IF (I).EQ.50)KK=INTV(I,J-1,I)-1
      IF (KK.LT.0)GO TO 10

```

C-TEST LOWER ELEMENT

```

C-      KK=INTV(I+1,J,I)
      IF (KK.LT.0)GO TO 10
      KTR=I+1
      LTR=I+1

```

C-LOAD THE NEW VALUE INTO ITS POSITION IN CODE

```

C-      IF (I+1)GO (SHIFT(K+1,LT),AND(AKT),SHIFT(HSK3,LT)))
      CONTINUE
      RETURN
      END

```

06/01/77 09.31.42

FTN 4.5414

SURROUTINE MVM1 74/74 OPT=1

```

1  SUBROUTINE MVM1(A, IFROM, R, ITO, K)
   C- THIS SUBROUTINE TRANSFERS K WORDS FROM CORP. BEGINNING WITH LOCATION A(IFROM).
   C- TO CORP. BEGINNING WITH LOCATION R(ITO).
   C-
      DIMENSION A(1), R(1)
      INTEGER A, R
      FORMAT(*, INSIDE MVM1 *, 3(5X, I10))
      I=IFROM
      J=ITO
      DO 10 L=1, K
         R(J)=A(I)
         I=I+1
         J=J+1
      RETURN
      END
10
15

```

```

1  PROGRAM PAPERLOT, OPT=1
2  C-THIS PROGRAM DRAWS THE TRACE PARTICLES ON THE PLOTTER AS ELLIPSES: USING AS
3  C-ONE OF THE OUTPUT TAPES OF PROGRAM GENP00, WHICH CONTAINS PARTICLE GEOMETRIC DATA
4  C-
5  C-
6  C-
7  C-
8  C-
9  C-
10 C-
11 C-
12 C-
13 C-
14 C-
15 C-
16 C-
17 C-
18 C-
19 C-
20 C-
21 C-
22 C-
23 C-
24 C-
25 C-
26 C-
27 C-
28 C-
29 C-
30 C-
31 C-
32 C-
33 C-
34 C-
35 C-
36 C-
37 C-
38 C-
39 C-
40 C-
41 C-
42 C-
43 C-
44 C-
45 C-
46 C-
47 C-
48 C-
49 C-
50 C-
51 C-
52 C-
53 C-
54 C-
55 C-
56 C-
57 C-
58 C-
59 C-
60 C-
61 C-
62 C-
63 C-
64 C-
65 C-
66 C-
67 C-
68 C-
69 C-
70 C-
71 C-
72 C-
73 C-
74 C-
75 C-
76 C-
77 C-
78 C-
79 C-
80 C-
81 C-
82 C-
83 C-
84 C-
85 C-
86 C-
87 C-
88 C-
89 C-
90 C-
91 C-
92 C-
93 C-
94 C-
95 C-
96 C-
97 C-
98 C-
99 C-
100 C-
101 C-
102 C-
103 C-
104 C-
105 C-
106 C-
107 C-
108 C-
109 C-
110 C-
111 C-
112 C-
113 C-
114 C-
115 C-
116 C-
117 C-
118 C-
119 C-
120 C-
121 C-
122 C-
123 C-
124 C-
125 C-
126 C-
127 C-
128 C-
129 C-
130 C-
131 C-
132 C-
133 C-
134 C-
135 C-
136 C-
137 C-
138 C-
139 C-
140 C-
141 C-
142 C-
143 C-
144 C-
145 C-
146 C-
147 C-
148 C-
149 C-
150 C-
151 C-
152 C-
153 C-
154 C-
155 C-
156 C-
157 C-
158 C-
159 C-
160 C-
161 C-
162 C-
163 C-
164 C-
165 C-
166 C-
167 C-
168 C-
169 C-
170 C-
171 C-
172 C-
173 C-
174 C-
175 C-
176 C-
177 C-
178 C-
179 C-
180 C-
181 C-
182 C-
183 C-
184 C-
185 C-
186 C-
187 C-
188 C-
189 C-
190 C-
191 C-
192 C-
193 C-
194 C-
195 C-
196 C-
197 C-
198 C-
199 C-
200 C-
201 C-
202 C-
203 C-
204 C-
205 C-
206 C-
207 C-
208 C-
209 C-
210 C-
211 C-
212 C-
213 C-
214 C-
215 C-
216 C-
217 C-
218 C-
219 C-
220 C-
221 C-
222 C-
223 C-
224 C-
225 C-
226 C-
227 C-
228 C-
229 C-
230 C-
231 C-
232 C-
233 C-
234 C-
235 C-
236 C-
237 C-
238 C-
239 C-
240 C-
241 C-
242 C-
243 C-
244 C-
245 C-
246 C-
247 C-
248 C-
249 C-
250 C-
251 C-
252 C-
253 C-
254 C-
255 C-
256 C-
257 C-
258 C-
259 C-
260 C-
261 C-
262 C-
263 C-
264 C-
265 C-
266 C-
267 C-
268 C-
269 C-
270 C-
271 C-
272 C-
273 C-
274 C-
275 C-
276 C-
277 C-
278 C-
279 C-
280 C-
281 C-
282 C-
283 C-
284 C-
285 C-
286 C-
287 C-
288 C-
289 C-
290 C-
291 C-
292 C-
293 C-
294 C-
295 C-
296 C-
297 C-
298 C-
299 C-
300 C-
301 C-
302 C-
303 C-
304 C-
305 C-
306 C-
307 C-
308 C-
309 C-
310 C-
311 C-
312 C-
313 C-
314 C-
315 C-
316 C-
317 C-
318 C-
319 C-
320 C-
321 C-
322 C-
323 C-
324 C-
325 C-
326 C-
327 C-
328 C-
329 C-
330 C-
331 C-
332 C-
333 C-
334 C-
335 C-
336 C-
337 C-
338 C-
339 C-
340 C-
341 C-
342 C-
343 C-
344 C-
345 C-
346 C-
347 C-
348 C-
349 C-
350 C-
351 C-
352 C-
353 C-
354 C-
355 C-
356 C-
357 C-
358 C-
359 C-
360 C-
361 C-
362 C-
363 C-
364 C-
365 C-
366 C-
367 C-
368 C-
369 C-
370 C-
371 C-
372 C-
373 C-
374 C-
375 C-
376 C-
377 C-
378 C-
379 C-
380 C-
381 C-
382 C-
383 C-
384 C-
385 C-
386 C-
387 C-
388 C-
389 C-
390 C-
391 C-
392 C-
393 C-
394 C-
395 C-
396 C-
397 C-
398 C-
399 C-
400 C-
401 C-
402 C-
403 C-
404 C-
405 C-
406 C-
407 C-
408 C-
409 C-
410 C-
411 C-
412 C-
413 C-
414 C-
415 C-
416 C-
417 C-
418 C-
419 C-
420 C-
421 C-
422 C-
423 C-
424 C-
425 C-
426 C-
427 C-
428 C-
429 C-
430 C-
431 C-
432 C-
433 C-
434 C-
435 C-
436 C-
437 C-
438 C-
439 C-
440 C-
441 C-
442 C-
443 C-
444 C-
445 C-
446 C-
447 C-
448 C-
449 C-
450 C-
451 C-
452 C-
453 C-
454 C-
455 C-
456 C-
457 C-
458 C-
459 C-
460 C-
461 C-
462 C-
463 C-
464 C-
465 C-
466 C-
467 C-
468 C-
469 C-
470 C-
471 C-
472 C-
473 C-
474 C-
475 C-
476 C-
477 C-
478 C-
479 C-
480 C-
481 C-
482 C-
483 C-
484 C-
485 C-
486 C-
487 C-
488 C-
489 C-
490 C-
491 C-
492 C-
493 C-
494 C-
495 C-
496 C-
497 C-
498 C-
499 C-
500 C-
501 C-
502 C-
503 C-
504 C-
505 C-
506 C-
507 C-
508 C-
509 C-
510 C-
511 C-
512 C-
513 C-
514 C-
515 C-
516 C-
517 C-
518 C-
519 C-
520 C-
521 C-
522 C-
523 C-
524 C-
525 C-
526 C-
527 C-
528 C-
529 C-
530 C-
531 C-
532 C-
533 C-
534 C-
535 C-
536 C-
537 C-
538 C-
539 C-
540 C-
541 C-
542 C-
543 C-
544 C-
545 C-
546 C-
547 C-
548 C-
549 C-
550 C-
551 C-
552 C-
553 C-
554 C-
555 C-
556 C-
557 C-
558 C-
559 C-
560 C-
561 C-
562 C-
563 C-
564 C-
565 C-
566 C-
567 C-
568 C-
569 C-
570 C-
571 C-
572 C-
573 C-
574 C-
575 C-
576 C-
577 C-
578 C-
579 C-
580 C-
581 C-
582 C-
583 C-
584 C-
585 C-
586 C-
587 C-
588 C-
589 C-
590 C-
591 C-
592 C-
593 C-
594 C-
595 C-
596 C-
597 C-
598 C-
599 C-
600 C-
601 C-
602 C-
603 C-
604 C-
605 C-
606 C-
607 C-
608 C-
609 C-
610 C-
611 C-
612 C-
613 C-
614 C-
615 C-
616 C-
617 C-
618 C-
619 C-
620 C-
621 C-
622 C-
623 C-
624 C-
625 C-
626 C-
627 C-
628 C-
629 C-
630 C-
631 C-
632 C-
633 C-
634 C-
635 C-
636 C-
637 C-
638 C-
639 C-
640 C-
641 C-
642 C-
643 C-
644 C-
645 C-
646 C-
647 C-
648 C-
649 C-
650 C-
651 C-
652 C-
653 C-
654 C-
655 C-
656 C-
657 C-
658 C-
659 C-
660 C-
661 C-
662 C-
663 C-
664 C-
665 C-
666 C-
667 C-
668 C-
669 C-
670 C-
671 C-
672 C-
673 C-
674 C-
675 C-
676 C-
677 C-
678 C-
679 C-
680 C-
681 C-
682 C-
683 C-
684 C-
685 C-
686 C-
687 C-
688 C-
689 C-
690 C-
691 C-
692 C-
693 C-
694 C-
695 C-
696 C-
697 C-
698 C-
699 C-
700 C-
701 C-
702 C-
703 C-
704 C-
705 C-
706 C-
707 C-
708 C-
709 C-
710 C-
711 C-
712 C-
713 C-
714 C-
715 C-
716 C-
717 C-
718 C-
719 C-
720 C-
721 C-
722 C-
723 C-
724 C-
725 C-
726 C-
727 C-
728 C-
729 C-
730 C-
731 C-
732 C-
733 C-
734 C-
735 C-
736 C-
737 C-
738 C-
739 C-
740 C-
741 C-
742 C-
743 C-
744 C-
745 C-
746 C-
747 C-
748 C-
749 C-
750 C-
751 C-
752 C-
753 C-
754 C-
755 C-
756 C-
757 C-
758 C-
759 C-
760 C-
761 C-
762 C-
763 C-
764 C-
765 C-
766 C-
767 C-
768 C-
769 C-
770 C-
771 C-
772 C-
773 C-
774 C-
775 C-
776 C-
777 C-
778 C-
779 C-
780 C-
781 C-
782 C-
783 C-
784 C-
785 C-
786 C-
787 C-
788 C-
789 C-
790 C-
791 C-
792 C-
793 C-
794 C-
795 C-
796 C-
797 C-
798 C-
799 C-
800 C-
801 C-
802 C-
803 C-
804 C-
805 C-
806 C-
807 C-
808 C-
809 C-
810 C-
811 C-
812 C-
813 C-
814 C-
815 C-
816 C-
817 C-
818 C-
819 C-
820 C-
821 C-
822 C-
823 C-
824 C-
825 C-
826 C-
827 C-
828 C-
829 C-
830 C-
831 C-
832 C-
833 C-
834 C-
835 C-
836 C-
837 C-
838 C-
839 C-
840 C-
841 C-
842 C-
843 C-
844 C-
845 C-
846 C-
847 C-
848 C-
849 C-
850 C-
851 C-
852 C-
853 C-
854 C-
855 C-
856 C-
857 C-
858 C-
859 C-
860 C-
861 C-
862 C-
863 C-
864 C-
865 C-
866 C-
867 C-
868 C-
869 C-
870 C-
871 C-
872 C-
873 C-
874 C-
875 C-
876 C-
877 C-
878 C-
879 C-
880 C-
881 C-
882 C-
883 C-
884 C-
885 C-
886 C-
887 C-
888 C-
889 C-
890 C-
891 C-
892 C-
893 C-
894 C-
895 C-
896 C-
897 C-
898 C-
899 C-
900 C-
901 C-
902 C-
903 C-
904 C-
905 C-
906 C-
907 C-
908 C-
909 C-
910 C-
911 C-
912 C-
913 C-
914 C-
915 C-
916 C-
917 C-
918 C-
919 C-
920 C-
921 C-
922 C-
923 C-
924 C-
925 C-
926 C-
927 C-
928 C-
929 C-
930 C-
931 C-
932 C-
933 C-
934 C-
935 C-
936 C-
937 C-
938 C-
939 C-
940 C-
941 C-
942 C-
943 C-
944 C-
945 C-
946 C-
947 C-
948 C-
949 C-
950 C-
951 C-
952 C-
953 C-
954 C-
955 C-
956 C-
957 C-
958 C-
959 C-
960 C-
961 C-
962 C-
963 C-
964 C-
965 C-
966 C-
967 C-
968 C-
969 C-
970 C-
971 C-
972 C-
973 C-
974 C-
975 C-
976 C-
977 C-
978 C-
979 C-
980 C-
981 C-
982 C-
983 C-
984 C-
985 C-
986 C-
987 C-
988 C-
989 C-
990 C-
991 C-
992 C-
993 C-
994 C-
995 C-
996 C-
997 C-
998 C-
999 C-
1000 C-

```

06/01/77 09.10.14

FV 4.5416

PROGRAM THRESH 7474 OPT=1

```

1  PROGRAM THRESH(INPUT,OUTPUT,TAPE1)
   DIMENSION IN(200),OUT(120)

5  C-THIS PROGRAM PRINTS THE THRESHOLD FOR IMAGE ON THE LINE PRINTED
   C-NRONS---LAST RECORD OF IMAGE TO BE PRINTED(ASSUMES 500)
   C-NRONS---LAST RECORD OF IMAGE TO BE PRINTED(ASSUMES 500)
   C-NRONS---LAST RECORD OF IMAGE TO BE PRINTED(ASSUMES 500)
   C-IPRM---STARTING RECORD OF IMAGE TO BE PRINTED(ASSUMES 1)
   C-ICOL---STARTING SAMPLE OF IMAGE TO BE PRINTED(ASSUMES 1)
10  C-
   NAMELIST/size/nrons,ncols,iprm,icol

15  C-
   C-ASSIGN DEFAULT VALUES TO INPUT PARAMETERS
   C-
   Nrons=ncols=500
   IPRM=ICOL=1
   PRINT 110
   READ SIZE
   PRINT SIZE
   NRP=(Ncols*2)/60
   IF((NRP*60)/2).LT.Ncols)NRP=NRP+1
   M=4
   DO 50 I=ICOL,NCOLS,120
     PRINT 120

25  C-
   C-SKIP FORWARD TO STARTING RECORD
   C-
   IF (IPRM.EQ.1) GO TO 20
   L=IPRM-1
   DO 10 I=1,L
     BUFFER IN(1,1) (IN(1),IN(2))
     IF UNIT(1) 10,5,10
10  CONTINUE
20  CONTINUE
   IR=((II-1)*2)/60+1
   IF (IR.GT.NRP)M=NR-IR+1
   DO 55 I=IPRM,NRONS
     BUFFER IN(1,1) (IN(1),IN(2))
     IF UNIT(1) 30,5,60
30  CONTINUE
   DO 30 J=1,M
     L=(J-1)*10+1
     LL=J*10
     KE=54
     JJ=IR+J-1
     DO 40 K=LL,LL
       IOUT(K)=MIN(SHIFT(IN(JJ),-KE),3)
40  CONTINUE
50  CONTINUE
     KE=KE-2
     MM=(M*60)/2
     PRINT 100,(IOUT(K),K=1,MM)
55  CONTINUE
   PRINT 1
60  CONTINUE
   STOP
   5  CONTINUE

```


06/01/77 09.37.51

FTN 4.5414

PROGRAM PICTURE 7/7/4 00741

139 00407414.13611
150 00407414.13611

81

201
200

```

1      PROGRAM FREQUPL(TINPUT,OUTPUT,TAPE2,PLFILE=0)
C- THIS PROGRAM USES THE OUTPUT OF PROGRAM HISTOCC AS INPUT (TAPE2) TO PRODUCE
C- PLOTS OF FREQUENCY VS. DENSITY FOR RAW IMAGE PIXELS
5      DIMENSION X(260),Y(260)
      READ *,NTIMES
      CALL COMPOS
      DO 70 K=1,NTIMES
10         I=0
          READ(2,101) IX,IY
          IF(EOF(2).NE.0) GO TO 10
          I=I+1
          X(I)=FLOAT(IX)
          Y(I)=FLOAT(IY)
          GO TO 20
15         CONTINUE
          CALL RGNDPL(J)
          CALL TITLE(17HREQ. VS. DENSITY,17.7HODENSITY,7.0HDEFQUENCY,0.0,0.0,
20          1.0,0.0)
          CALL GSAF(0.,30.0,255.0,0.0,0.0,0.0,0.0,250.00)
          CALL CURVE(X,Y,I,0)
          CALL ENDPL(J)
          STOP
25         FORMAT(10,I11)
          ENDO
100      END

```


PAGE 2

06/01/77 09.14.20

FTN 4.54414

PROGRAM SAMPLET 74/74 OPT=1

120 FORMAT(F4.2,X,F6.2)
130 FORMAT(1H ,F8.2,X,F6.2)
END

60

APPENDIX G
DIGIPLT USER'S GUIDE

I. INTRODUCTION

GENERAL DESCRIPTION

DIGIPLT is an interactive data curve processor; a collection of programs that enables its user to enter, display, edit, and operate on data curves. It is an easy-to-use software package, and does not require that the user have extensive knowledge of computer processing.

While designed primarily for use on the TEKTRONIX 4014 graphics terminal, the user may take advantage of many DIGIPLT capabilities using any interactive computer terminal on-line with the ASD CDC-6600 computer in INTERCOM mode.

The user should note that a brief synopsis of this guide may be obtained at the initial stage of DIGIPLT execution by responding "YES" when asked by DIGIPLT if help is desired.

DIGIPLT is, basically, a collection of programs that are executed upon user request. The request to execute a particular program is made when the user specifies one of the five-character codes on the DIGIPLT option menu. Execution of each of the programs on the menu is completely independent of the rest; thus, the user may select menu options in any order desired.

RESPONSE TO DIGIPLT REQUESTS/QUESTIONS

All of the keyboard entries the user is required to make during DIGIPLT execution are preceded by a printed message (output by DIGIPLT) on the terminal screen specifying the input expected. The response to DIGIPLT requests must be an integer, or a "YES" or "NO," depending upon the nature of the request. There are three cases, however, when they must be real numbers.

1. When entering axis information in DGTZR.
2. When entering new X and Y values when adding or replacing points in EDTAL.
3. When entering X and Y values when creating a data file in CRTFL.

If the request is for a "YES" or "NO" response, entering only a "Y" will not suffice for a "YES," as will an "N" not suffice for a "NO." The user must enter "YES" or "NO." In most cases, any other input will be treated as a "NO" response.

II. DIGIPLT DATA FILE STRUCTURE

The purpose of this section is to describe the data files used by DIGIPLT. This information would be of use to all DIGIPLT users, as it is essential to be familiar with the data files used by DIGIPLT, since all data processed by DIGIPLT must exist on a DIGIPLT data file in the correct format, and almost all processing is done in terms of these files.

DIGIPLT DATA FILES

The user has available five separate files on which to store data. Data is placed on these files by one of two methods: 1) the user has, prior to execution, created a data file which is compatible with the DIGIPLT data file structure, or 2) the user creates a data file through the use of the DIGIPLT options DGTZR and/or CRTFL. In the latter case, the data file is automatically formatted in DIGIPLT data file structure. The user directs DIGIPLT to use a specific data file by entering its "file number" ("data file number") when requested to do so by DIGIPLT. At all times during execution of DIGIPLT, data files are referred to by their file numbers. These file numbers (integers) correspond to actual mass storage files as follows:

<u>DIGIPLT FILE NO.</u>	<u>ACTUAL MASS STORAGE FILE NAME</u>
1	Tape 1
2	Tape 2
3	Tape 3
4	Tape 4
5	Tape 5

Only those files listed above are valid DIGIPLT data files. Any attempt to process other files will result in an error. Data files created prior to DIGIPLT execution should be on a valid mass storage file, and be referred to during DIGIPLT execution by its corresponding file number. Similarly, files created by DIGIPLT should be referred to by their actual mass storage file name when used after completion of DIGIPLT execution.

RECORD FORMAT

Data exists on DIGIPLT data files as two real numbers per record, in free-format (separated by a blank or a comma). Each record corresponds to one data point; the first number on the record being the abscissa value, and the second number the ordinate value of that point (X, Y).

SPECIAL RECORDS (FLAGS)

There are three types of special records on DIGIPLT data files: 1) end-of-block (EOB) flag, also called end-of-curve flags, 2) end-of-data (EOD) flag, also called end-of-file (EOF) flag, and 3) header block records.

1. END-OF-BLOCK (EOB) Flag - This is a record having values of "9999.0" for both its X and Y positions. It is used to indicate

the end of a block of data, where a block of data is treated by DIGIPLT as one data (exception - see header block description) curve.

2. END-OF-DATA Flag - The last record in the data file, which has both an X and Y value of "8888.0." It should be noted that for the last block on the file there will be no EOB flag; instead, the EOD flag (EOF flag) is located in the last record. There must be only one EOD flag on a file; any data after an EOD flag on a DIGIPLT data file will be ignored and subsequently destroyed. If the EOD flag is not present, an error will occur when attempting to process the data file.
3. HEADER BLOCK RECORDS - The first block on all DIGIPLT data files contains information on the contents of the file and is used by DIGIPLT for scaling purposes. This block is three records (points) in length, and contains the following:

<u>REC</u>	<u>X POSITION</u>	<u>Y POSITION</u>
1	XMAX	YMAX
2	XMIN	YMIN
3	EOB	

where: XMAX and YMAX are the maximum X and Y values, respectively, of the data points on the file; XMIN and YMIN are the minimum X and Y values, respectively, of the data points on the file.

It should be noted that, due to the header block, the first actual data curve on a file is actually the second data block. The user should keep this in mind when DIGIPLT asks for "BLOCK NO." and "CURVE NO." from the user - a data curve's "BLOCK NO." will always be one greater than its "CURVE NO."

PERMANENT FILES

If the user desires to use data located on a permanent file cataloged prior to DIGIPLT execution, he must copy the file to one of the DIGIPLT data files before execution of DIGIPLT. If a permanent file is attached as one of the DIGIPLT data files, an unrecoverable error will occur in DIGIPLT execution due to the fact that DIGIPLT writes on the data files it processes, which is an illegal operation on a permanent file. The user may, however, request that a DIGIPLT data file be assigned to a permanent file device and catalog that file after DIGIPLT execution.

III. MENU OPTIONS

The following section elaborates on each of the user-selectable programs of DIGIPLT (menu options). A menu option is executed when the user enters its 5-character option code in response to the DIGIPLT query "ENTER 5-CHAR. OPTION CODE."

ALPHA-NUMERIC DATA CURVE DISPLAY (DSPAL)

The function of DSPAL is to produce a listing of the abscissa and ordinate values of points on specific data blocks. This display will indicate the block number, point number, and X and Y values for all blocks specified to be displayed on a given data file.

Since DSPAL displays are in terms of blocks, the user may display the header block of each file. Therefore, the first actual data curve on the file will be block no. 2. When asked to input the block numbers to be displayed, an entry of "0" for the first block number will cause all blocks on the specified data file to be displayed. If the user chooses to display only certain blocks, they should be entered in ascending order by their block numbers. A "0" must be entered as the last block to be displayed.

EXAMPLE OF DSPAL EXECUTION.
 HERE, A "0" WAS ENTERED AS THE
 FIRST BLOCK TO BE DISPLAYED,
 RESULTING IN THE ENTIRE FILE
 CONTENTS BEING DISPLAYED.

DSPAL
 ENTER NO. OF FILE CONTAINING DATA
 1
 DIGIT 1.0 DSPAL LEVEL
 ENTER BLOCKS TO BE DISPLAYED IN
 INCREASING ORDER. ENTER NO. '0'
 FOR LAST BLOCK
 BLOCK NO. 1 - 0

BLOCK	PT. NO.	X	Y
1	1	10.00	10.00
1	2	0.00	0.00
1	3	9999.00	9999.00

BLOCK	PT. NO.	X	Y
2	1	0.00	0.00
2	2	1.00	1.00
2	3	2.00	2.00
2	4	3.00	3.00
2	5	4.00	4.00
2	6	5.00	5.00
2	7	10.00	10.00
2	8	9999.00	9999.00

BLOCK	PT. NO.	X	Y
3	1	0.00	0.00
3	2	10.00	0.00
3	3	10.00	10.00
3	4	0.00	10.00
3	5	0.00	0.00
3	6	8882.00	8888.00

TYPE IN 5-CHAR. FUNCTION NAME

IN THIS EXAMPLE OF DSPAL
EXECUTION, ONLY THE 2ND AND
3RD BLOCKS OF THE DATA FILE
WERE DISPLAYED.

DSPAL
ENTER NO. OF FILE CONTAINING DATA
1
DISPLAY 1-9 DSPAL LEVEL
ENTER BLOCKS TO BE DISPLAYED IN
ORDER, IN ORDER. ENTER NO. 10.
001 1ST BLOCK
BLOCK NO. 1-2
BLOCK NO. 2-3
BLOCK NO. 3-6

BLOCK	PT. NO.	X	Y
1	1	0.00	0.00
1	2	1.00	1.00
1	3	2.00	2.00
1	4	3.00	3.00
1	5	4.00	4.00
1	6	5.00	5.00
1	7	10.00	10.00
1	8	9999.00	9999.00

BLOCK	PT. NO.	X	Y
3	1	0.00	0.00
3	2	10.00	0.00
3	3	10.00	10.00
3	4	0.00	10.00
3	5	0.00	0.00
3	6	8888.00	8888.00

TYPE IN 5-CHAR. FUNCTION NAME

GRAPHIC DATA FILE DISPLAY (DSPGR)

The function of the DSPGR option is to produce a plot on the terminal screen of specified data curves on a specific data file. The plot produced is a simply X-Y graph of the data curves, and enables the user to examine the data in graphic form.

Since DSPAL displays are in terms of data curves, a "1," for example, entered as the data curve to be displayed will result in a plot of the first actual data curve on the specified file. As in DSPAL, a "0" entered for the first data curve to be displayed will result in the display of all data curves on the file. A "0" must be entered as the last curve to be displayed.

The criteria for scaling of the displayed graph is determined by the method in which the user chooses to display the data curves. If he elects to display all curves on a file by entering a "0" as the first curve to be displayed, scaling of the graph will be in terms of the information in the header block of the data file. If a specific curve(s) to be displayed is entered, scaling of the graph will take place in relation to the minimum and maximum abscissa and ordinate values of the data to be displayed.

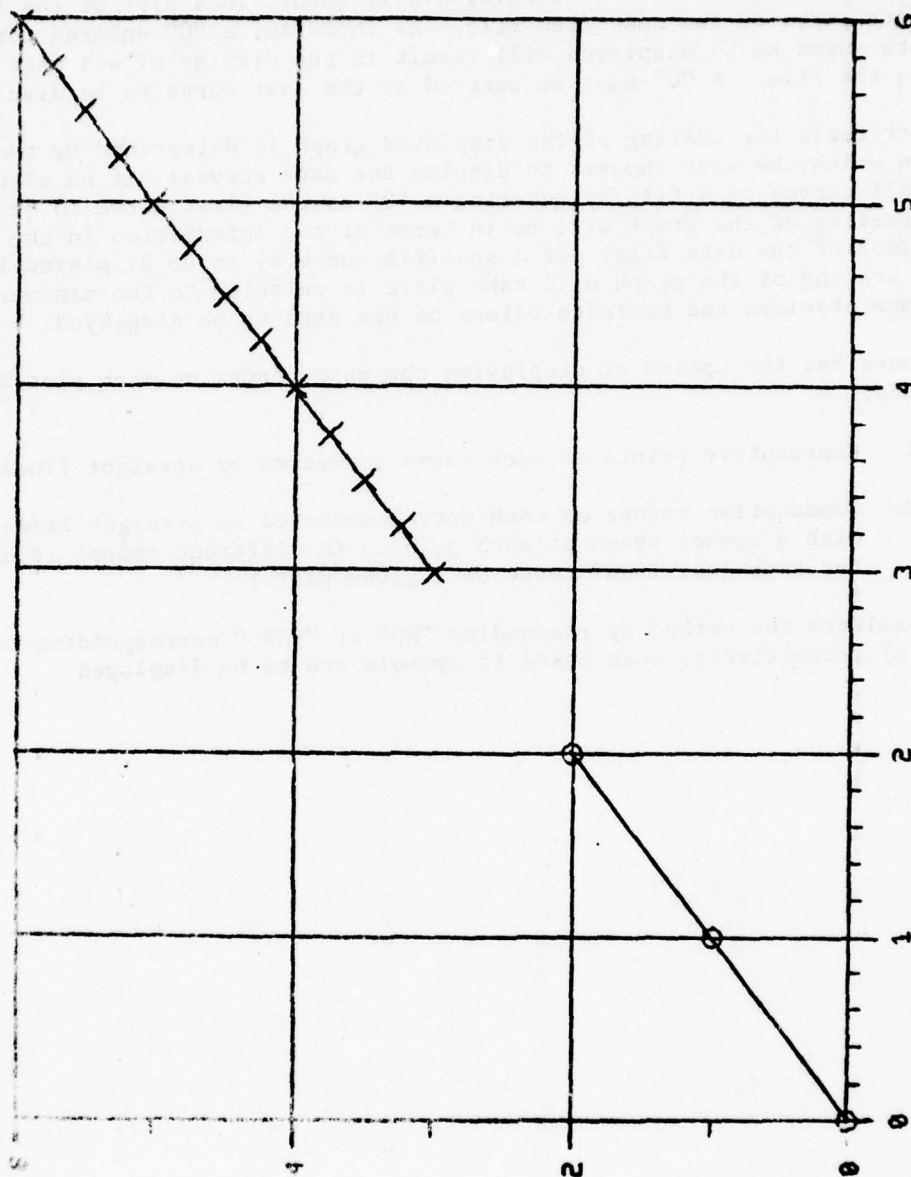
The user has the option of displaying the data curves on each plot by two methods:

1. Consecutive points on each curve connected by straight lines.
2. Consecutive points on each curve connected by straight lines, with a symbol drawn at each point. (A different symbol is used for each individual curve on a given plot.)

The user selects the method by responding "NO" or "YES," corresponding to (1) and (2) respectively, when asked if symbols are to be displayed.

DO YOU WANT VARIABLES DISPLAYED ON CURVES?

YES



EXAMPLE OF DSP6R OUTPUT. DATA PLOTTED HERE IS DATA
USED IN DSP6R EXAMPLE IN THIS GUIDE.

ALPHA-NUMERIC DATA FILE EDITOR (EDTAL)

The purpose of EDTAL is to provide the user with the capability of editing data blocks. As in DSPAL, EDTAL refers to the data file in terms of blocks, thus allowing the editing of the header block on a file.

The user has three editing functions available: 1) addition of new points to, 2) deletion of points from, and 3) replacing of points with new ones on a data block. These functions are effected by entering a one-letter edit command when asked by DSPAL to do so. Valid edit commands are as follows:

<u>1-Letter Command</u>	<u>Function</u>
A	Add point(s) to specified block
D	Delete point(s) from specified block
R	Replace point(s) on a specified block

The user is asked to enter the point number to be processed by EDTAL. This point number entered here should be the point number on the display corresponding to the position the user wishes to begin editing. In the case of deleting or replacing points, the point specified will actually be affected (example: point no. specified is 5, edit command was D, point no. 5 will be deleted). In the addition of point(s), the new point(s) will be inserted in the block immediately after the point number specified.

Both before and after executing an edit command, the user has the option of displaying the block he wishes to edit. All "point numbers" referred to in this discussion are those which appear on this display. (This display is of the same nature as that produced by DSPAL.)

When adding or replacing points, the user must enter the new X and Y values for each point. They must be entered as real numbers, separated by a blank or a comma.

The user, when specifying a deletion, has the option of deleting the entire block. If a "YES" response is given when the user asks if he wishes this done, the entire block will disappear from the data file. Note that if the last block of the file is deleted, the last record (end-of-block flag) of the preceding block must be changed from "9999.0 . 9999.0" to "8888.0 . 8888.0" to assure normal processing of that file and prevent the occurrence of errors. In the same respect, the user may add an entire block to the file by adding points after the end of block or end of data (file) flag, specifying as the last X-Y pair the end-of-block or end-of-data flag accordingly. In determining proper use of the above flags, it is important to remember that any data appearing after an end-of-data flag ("8888.0 . 8888.0") will be ignored and consequently deleted from the data file.

EXAMPLE OF EDTAL USE, HERE A
POINT WITH COORDINATES (6.0, 6.0) WAS
INSERTED AFTER THE 3rd POINT IN
THE 3rd BLOCK BY USING THE ADD (A)
COMMAND.

PER NO. OF FILE CONTAINING DATA

DISPLAY 1.0 EDTAL LEVEL

ENTER BLOCK NO. TO BE EDITED

DO YOU WANT THE BLOCK CONTENTS DISPLAYED?

YES

BLOCK	PT. NO.	X	Y
1	1	3.00	3.00
2	2	4.00	4.00
3	3	5.00	5.00
4	4	8888.00	8888.00

ENTER 4 : LETTER EDIT COMMAND

FOR RECORD NO. TO BE PROCESSED

DO YOU WANT THE BLOCK CONTENTS DISPLAYED?

YES

DO YOU WANT THE BLOCK CONTENTS DISPLAYED?

YES

BLOCK	PT. NO.	X	Y
1	1	3.00	3.00
2	2	4.00	4.00
3	3	5.00	5.00
4	4	6.00	6.00
5	5	8888.00	8888.00

DO YOU WANT TO EDIT MORE IN CURRENT BLOCK?

NO

DO YOU WANT TO EDIT ANOTHER BLOCK?

NO



ENTER NO. OF FILE CONTAINING DATA
2

DIGIPLT 1.0 EDTAL LEVEL
 ENTER BLOCK NO. TO BE EDITED
3

DO YOU WANT THE BLOCK CONTENTS DISPLAYED?
YES

BLOCK	PT. NO.	X	Y
3	1	3.00	3.00
3	2	4.00	4.00
3	3	5.00	5.00
3	4	8888.00	8888.00

ENTER A 1 LETTER EDIT COMMAND
D

DO YOU WANT TO DELETE THE ENTIRE BLOCK?
NO

ENTER RECORD NO. TO BE PROCESSED
3

HOW MANY POINTS DO YOU WISH TO ADD/REPLACE/DELETE?
1

DO YOU WANT THE BLOCK CONTENTS DISPLAYED?
YES

BLOCK	PT. NO.	X	Y
3	1	3.00	3.00
3	2	4.00	4.00
3	3	8888.00	8888.00

DO YOU WANT TO EDIT MORE IN CURRENT BLOCK?
YES

ENTER A 1 LETTER EDIT COMMAND
A

ENTER RECORD NO. TO BE PROCESSED
2

HOW MANY POINTS DO YOU WISH TO ADD/REPLACE/DELETE?
1

ENTER X AND Y VALUES, SEPARATED BY A BLANK OR A COMMA, FOR THE POINT NO.5 SPECIFIED
3- 5- 5.

DO YOU WANT THE BLOCK CONTENTS DISPLAYED?

YES
 BLOCK PT. NO. X Y

3	1	3.00	3.00
3	2	4.00	4.00
3	3	5.00	5.00
3	4	2888.00	2823.00

DO YOU WANT TO EDIT MORE IN CURRENT BLOCK?
NO

DO YOU WANT TO EDIT ANOTHER BLOCK?
NO

TYPE IN 5-CHAR. FUNCTION NAME

*IN THIS EXAMPLE OF EDTAL USE,
 THE 3RD POINT OF THE 3RD BLOCK
 WAS REMOVED USING THE DELETE (D)
 COMMAND; AND THEN PUT BACK USING
 THE ADD (A) COMMAND.*

ENTER NO. OF FILE CONTAINING DATA
2

REPLY 10 LEVEL

ENTER BLOCK NO. TO BE EDITED
2

DO YOU WANT THE BLOCK CONTENTS DISPLAYED?
YES

BLOCK	PT. NO.	X	Y
1	0.00	0.00	0.00
2	1.00	1.00	1.00
3	2.00	2.00	2.00
4	8888.00	8888.00	8888.00

ENTER A 1 LETTER EDIT COMMAND
2

ENTER RECORD NO. TO BE PROCESSED
2

DO YOU WISH TO ADD/REPLACE/DELETE?

ENTER X AND Y VALUES, SEPARATED BY A BLANK
OR A COMMA, FOR THE POINT NO.S SPECIFIED

4- 9999. 9999.

DO YOU WANT THE BLOCK CONTENTS DISPLAYED?
YES

BLOCK	PT. NO.	X	Y
1	0.00	0.00	0.00
2	1.00	1.00	1.00
3	2.00	2.00	2.00
4	9999.00	9999.00	9999.00

DO YOU WANT TO EDIT MORE IN CURRENT BLOCK?
YES

ENTER A 1 LETTER EDIT COMMAND
A

ENTER RECORD NO. TO BE PROCESSED
4

HOW MANY POINTS DO YOU WISH TO ADD/REPLACE/DELETE?
1

ENTER X AND Y VALUES, SEPARATED BY A BLANK
OR A COMMA, FOR THE POINT NO.S SPECIFIED

5- 3. 3.
6- 4. 4.
7- 5. 5.
8- 8888. 8888.

DO YOU WANT THE BLOCK CONTENTS DISPLAYED?
YES

BLOCK	PT. NO.	X	Y
1	0.00	0.00	0.00
2	1.00	1.00	1.00
3	2.00	2.00	2.00
4	9999.00	9999.00	9999.00
5	3.00	3.00	3.00
6	4.00	4.00	4.00
7	5.00	5.00	5.00
8	8888.00	8888.00	8888.00

DO YOU WANT TO EDIT MORE IN CURRENT BLOCK?
NO

DO YOU WANT TO EDIT ANOTHER BLOCK?
NO

TYPE IN 5-CHAR. FUNCTION NAME

IN THIS EXAMPLE, AN ADDITIONAL

BLOCK (CURVE) WAS ADDED TO THE END OF
DATA FILE BY 1) REPLACING THE EOD FLAG
(8888. 8888.) BY AN EOB FLAG (9999. 9999.)

AND ADDING THE POINTS OF THE NEW
CURVE, WITH AN EOD FLAG AS THE LAST
POINT.

DATA CURVE INTERPOLATION (INTRP)

The function of the INTRP option is to interpolate points between the existing points on a specified data curve(s). As in DSPAL and DSPGR, a "0" entered for the first curve to be processed will result in interpolation on all curves on the specified data file; if specific curve(s) are to be processed, a "0" must be entered as the last curve to be interpolated.

The user is asked to enter the number of intervals to be interpolated between the original points on the data curve(s). In order for interpolation to occur, this number must be in the range of 2 - 10 inclusive. INTRP will interpolate and insert between the original points of the data curve(s) the number of points equal to one less than the number of intervals specified. (Thus, if an interval of "1" was specified, no points would be interpolated.)

Due to limited storage space in the computer, individual data curves are limited to having a maximum of 900 points each. Should the number of intervals specified result in a curve of length greater than 900 points, the largest interval which will contain the curve to the maximum 900 points is automatically used by INTRP.

In the actual interpolation, INTRP uses a 3d degree, piece-wise fitting of the original data points to determine the value of interpolated points. The abscissa values of the interpolated points are determined by the number of intervals specified and the abscissa values of the original points; their ordinate values are determined by the abscissa values and the fit. It is mandatory that any curve being interpolated be continuous, and the abscissa values be increasing as their point numbers increase.

DATA FILE STATISTICS AND COPYING (FILES)

The function of the FILES option is to provide the user with the status of each of the five data files at his disposal, and enable the copying of one data file to another.

The number of data curves, the number of points on each curve, and the total number of points contained in the file are given for each file. If a file contains no data, the message "FILE EMPTY" is given for that file.

In copying files, any valid data file may be copied to any other valid data file. If the user attempts any copy involving non-existent files, an error message is printed and no copying occurs. If a file that is specified to be copied to contains data prior to the copy, a warning is given; and if the user elects to proceed with the copy, any data previously on the file will be destroyed.

FILES
RECEIVED 1.0 FILES LEVEL
OTHERWISE, TYPE ANYTHING
U
TYPE IN 5-CHAR. FUNCTION NAME

EXAMPLE OF FILES EXECUTION.
HERE DATA FILE #1 WAS COPIED TO
DATA FILE #3.

FILES
RECEIVED 1.0 FILES LEVEL

FILE TAPE1 STATISTICS
FILE NO. 1 ----- 2 DATA POINTS
FILE NO. 2 ----- 3 DATA POINTS
FILE NO. 3 ----- 13 DATA POINTS
FILE NO. 18 TOTAL POINTS

FILE TAPE2 STATISTICS
FILE NO. 1 ----- 2 DATA POINTS
FILE NO. 2 ----- 3 DATA POINTS
FILE NO. 3 ----- 13 DATA POINTS
FILE NO. 18 TOTAL POINTS

FILE TAPE3 STATISTICS
FILE EMPTY

FILE TAPE4 STATISTICS
FILE EMPTY

FILE TAPE5 STATISTICS
FILE EMPTY

ALL DATA FILES HAVE BEEN REWOUND
IF YOU WISH TO COPY ONE FILE TO ANOTHER, TYPE 'COPY'.
COPY
ENTER NO. OF FILE TO BE COPIED
1

ENTER NO. OF FILE TAPE1 IS TO BE COPIED TO
3
COPY COMPLETED

ALL DATA FILES HAVE BEEN REWOUND
IF YOU WISH TO COPY ONE FILE TO ANOTHER, TYPE 'COPY'.

DATA FILE CREATION FROM KEYBOARD ENTRY (CRTFL)

The purpose of the CRTFL option is to provide the user with the capability of creating data curve(s) by entering X-Y values from the terminal keyboard. Data curves may be created on any valid DIGIPLT data file. (A warning will be issued if an attempt is made to create data curves on a file currently containing data.)

The process of the creation is a point-by-point entry of data curves. An end of block flag entered (9999.0 . 9999.0) as the X and Y values results in the beginning of a new curve on the file. The last point entered must be an end-of-data flag ("8888.0 . 8888.0"); this results in the completion of the data file creation. CRTFL will continue to ask for X-Y values until the end-of-data flag is entered. An end-of-block flag must not be entered on the last block to be created. Each point must be entered as an X and a Y value, separated by a blank or comma, on the same line.

The header block required on all DIGIPLT data files is automatically created and placed in the correct location on the file created.

IN THIS EXAMPLE OF CRTFL USE,
DATA FILE #1 WAS CREATED FROM
KEYBOARD ENTRY OF POINTS.

DIGIT 1.0 DATA LEVEL
DO YOU NEED HELP?
NO
DIGIT 1.0 DATA LEVEL
TYPE IN 5-CHAR. FUNCTION NAME
CRTFL
DIGIT 1.0 CRTFL LEVEL
ENTER NO. OF FILE TO BE CREATED
1

ENTER IN X VALUE AND Y VALUE (MUST BE READ NO.5)
ON EACH LINE, SEPARATED BY A BLANK OR A COMMA
THE X AND Y VALUES OF THE LAST POINT ON THE FILE
POINT 1.0. 388.0. ENTER X AND Y VALUES OF '9999.0'
TO INDICATE THE BEGINNING OF A NEW DATA CURVE
DATA CURVE NO. 1 X Y
1.0 0.
2.0 1.
3.0 2.
4.0 3.
5.0 4.
6.0 5.
7.0 10.
8.0 999. 9999.

DATA CURVE NO. 2 X Y
POINT NO. X Y
1.0 0.
2.0 0.
3.0 10.
4.0 10.
5.0 0.
6.0 888. 8888.

DATA FILE TAPE 1
NO. OF CURVES = 2 TOTAL NO. OF POINTS = 12
TYPE IN 5-CHAR. FUNCTION NAME

DIGITIZER INPUT OPTION (DGTZR)

This option should be executed only if using a TEKTRONIX 4014 graphics terminal, with a TEKTRONIX 4954 digitizing tablet. Any mention of the "digitizer" in the following discussion refer to the 4954 tablet and controller.

TABLET OPERATION

This option provides the user with the capability of entering data curves by digitizing graphs. If it is intended to use this option, it is advisable to turn on the tablet controller before beginning execution of DIGIPLT.

Secure the graph to be digitized to the tablet with four small pieces of adhesive tape. It is advisable to place the graph near to the center of the tablet, as the usable area of the tablet does not necessarily extend to its edges. DIGIPLT automatically corrects for skew of the graph with respect to the tablet. While digitizing do not lift the mouse from the tablet. If this happens while the digitizer is armed (ready light on controller is on), the ready status of the digitizer will be lost. In this case, return the mouse to the tablet and move it around until the ready status returns. Do not key the mouse unless the tablet is armed. To digitize a point, place the crosshairs of the mouse over the point, and key the mouse once. Do not move the mouse while it is keyed. DIGIPLT automatically arms the tablet when it expects input from the digitizer.

GRAPH FORMAT

Any graph that is to be digitized must have an X (abscissa) axis, and a Y (ordinate) axis perpendicular with respect to the X axis, associated with the graph (i.e. must have a cartesian coordinate system).

It is also required that there be associated with these axes:

1. an abscissa and ordinate value at the origin.
2. an abscissa value at the maximum extent (pos. sense) of the X axis.
3. an ordinate value at the maximum extent (pos. sense) of the Y axis.

These values may be arbitrary. The points digitized will have coordinates valued with respect to the above. To establish the relation between the values of the points above to the coordinate values output by the tablet, DGTZR will ask the user to type in the values, (1, 2, and 3 above) and then digitize the corresponding points on the graph.

At this point, DGTZR will ask the user to input (digitize) the curve with the mouse, entering a "D" from the keyboard when finished digitizing that curve. When entering a curve, move the mouse along the desired curve, keying it at the locations where actual points are to be taken. Begin at the left-most point on the curve and proceed to its end. The output of the digitizer is echoed on the terminal screen (appears as "strange" character

strings). If many points are being digitized, be careful that the terminal screen does not become "full," causing the terminal to stop receiving data from the digitizer. When the curve has been digitized, type a "D" and a carriage return on the keyboard. If at this time there is another curve to be digitized, respond "YES" when asked by DGTZR if another curve is to be input. A "NO" response here will result in final processing of the data entered and exit from DGTZR.

EXIT FROM MENU OPTION SELECTION MODE (FINSH)

The purpose of the FINSH option is to exit from the user-selectable option mode. This option is executed when the user has completed processing all data. Upon exiting, DIGIPLT will ask the user if he is finished. A "YES" reply given at this point will result in the termination of DIGIPLT execution. All data will remain on the DIGIPLT data files, for the user to do with as he wishes (example: catalog as permanent file, dispose to a batch terminal, or use as input to another program). A "NO" response will result in the return of the user-selectable option mode, and the user may continue processing.

IV. USE OF DIGIPLT FROM TERMINAL OTHER THAN TEKTRONIX 4014

DIGIPLT, while primarily designed to be executed using a TEKTRONIX 4014 terminal, can be executed using any terminal on-line with the CDC-6600 (INTERCOM). Two restrictions are observed:

1. DO NOT SELECT DGTZR OPTION.
2. DO NOT SELECT DSPGR OPTION.

The procedures for use of all DIGIPLT options other than those above from a terminal other than the TEKTRONIX 4014 are exactly as outlined in this guide.

V. GENERAL INFORMATION

EXECUTION OF DIGIPLT

The DIGIPLT program is maintained in absolute binary form on the AFML program library. To execute DIGIPLT, the user must login to INTERCOM on the CDC-6600, attach DIGIPLT from the library, and execute. Since DIGIPLT is in absolute form it requires no additional supporting software.

DIGIPLT is attached and executed by the following commands:

```
ATTACH,DIGIPLT,DIGIPLT,ID=M754321.  
DIGIPLT.
```

These commands must be given exactly as shown. Any variations will result in execution errors.

The user should become, if not already, familiar with the use of CDC-6600 INTERCOM. Information on its use is available from AFML/DOC.

IF DIGIPLT BOMBS

Should the user cause DIGIPLT to bomb, all data files are preserved in the state they were in immediately before the termination of execution. INTERCOM commands may be used to examine data (ex.: COPYSBF, TAPE1, OUTPUT) attempt to ascertain the cause of the error, if not known already. The user may elect to use DIGIPLT to examine the data. In this case, it is strongly advisable to select the FILES option before any other, as this will give the user the most information on the data files.

REPORTING BUGS AND ERRORS

Should the user discover a bug or error in DIGIPLT, he is asked to document and report the problem to AFML/DOC. The documentation need only be the terminal output sufficient to show the existence of the bug or error, or the equivalent explanation.

ADDITIONAL INFORMATION

Manuals on the use of INTERCOM, the TEKTRONIX 4014 terminal, and TEKTRONIX 4954 digitizing tablet are available at AFML/DOC, should the user desire more information on these.

VI. SPECIFICATIONS

EXECUTION REQUIREMENTS

Central Memory 60000₈ words (60-bit)
Time Dependent upon length of "session" of use
Mode Interactive. Must use CDC-6600 INTERCOM
File Requirements Uses Mass Storage files:

Tape 1
Tape 2
Tape 3
Tape 4
Tape 5
Tape 99

PROCESSING LIMITATIONS

Maximum No. of Data Curves/File 50
Maximum No. of Data Points/Curve 900

APPENDIX H
GENERAL USER PROGRAMS FOR THE CDC 6600

```

1  C-OPEN MICROM PROCESES ELECTRON MICROPORE DATA: RE=FORMATTING AND
2  C-CALCULATING THE MEAN AND STANDARD DEVIATION.
3  C-
4  C-INPUT
5  C-
6  C-INPUT DATA IS IN THE FORM OF PUNCHED CARDS. THESE CARDS ARE GENERATED
7  C- BY CONVERTING THE PAPER TAPE OUTPUT OF THE MICROPORE TO PUNCHED CARDS.
8  C- THE 1ST CARD IN EACH DATA SET MUST HAVE AN 'E' IN THE 2ND COLUMN
9  C- (MULTIPLE DATA SETS MAY BE PROCESSED USING 1 RUN).
10 C-
11 C-OUTPUT
12 C-
13 C-OUTPUT IS THE RE-FORMATTED DATA (F7.0) ON MASS STORAGE FILE TAPE2 AND
14 C- LINE PRINTER: AND POINTED MEAN AND STANDARD DEVIATION.
15 C-
16 C- DIMENSION X(1000)
17 C-
18 C-
19 C-
20 C-READ DATA CARDS: TEST FOR VALIDITY (IFLAG=1H2). END-OF-DATA SET (IFLAG = 1HE). AND
21 C- END OF FILE (PRINT) DATA VALUES
22 C-
23 10 CONTINUE
24 READ 100, IFLAG, X(I)
25 IF (COF(5LINPUT)) 50, 20
26 CONTINUE
27 IF (IFLAG.EQ.1H2) GO TO 30
28 IF (IFLAG.EQ.1HF) GO TO 50
29
30 C-PRINT ERROR MESSAGE FOR INVALID CARD
31 C-
32 PRINT 110, I
33 GO TO 10
34 PRINT 120, I, X(I)
35 I=I+1
36 GO TO 10
37 CONTINUE
38 IF (I.NE.1) GO TO 50
39 PRINT 130
40 STOP
41 CONTINUE
42 I=I-1
43
44 C-WRITE DATA ON FILE TAPE2
45 C-
46 70 DO 70 J=1, I
47 WRITE (2, 140) X(J)
48 ENDTLF 2
49 REWIND 2
50
51 C-READ DATA IN FLOATING POINT FORMAT FROM TAPE2 FOR CALCULATIONS
52 C-
53 90 DO 90 J=1, I
54 READ (2, 150) X(J)
55 IF (COF(2)) 85, 90
56 CONTINUE
57 95 CONTINUE

```

169

05/01/77 09.31.23

FTN 4.5+414

SUBROUTINE STAY 74/74 OPT=1

```

1  C-CALCULATE THE STANDARD DEVIATION
2  C-THIS SUBROUTINE CALCULATES THE MEAN (XM) AND STANDARD DEVIATION (S) FOR A
3  C-SPRING OF N NUMBERS IN THE ARRAY X.
4  C-
5  C-
6  C-
7  C-
8  C-
9  C-
10 C-CALCULATE THE MEAN
11 C-
12 C-
13 DO 10 I=1,N
14   XM=XM+X(I)
15   XM=XM/FLOAT(N)
16 C-CALCULATE THE STANDARD DEVIATION
17 C-
18 C-
19 DO 20 I=1,N
20   XSQ=XSQ+(X(I)-XM)**2
21   S=SQRT(XSQ/FLOAT(N-1))
22 RETURN
23 END

```


06/01/77 09.19.24

FTN 6.5414

PROGRAM WALT 74/74 OPT=1

```

1  PROGRAM WALT(INPUT,OUTPUT,OLFILE=0)
   DIMENSION X(9,7)
   CALL COMPRS
   N=1

5  C-
   C-THE " I " LOOP IS A COUNTDOWN FOR THE 9-CAPD DATA SETS
   C-
   DO 20 I=1,4
   C-
10  C-READ IN DATA FOR 3 PLOTS(6 CURVES)
   C-
   DO 10 J=1,9
   READ 100,(X(J,L),L=1,7)
   C-
15  C-
   C-SET UP AND PLOT 3 CURVES
   C-
   DO 5 M=1,3
   CALL RGNPL(N)
   CALL PAGE(10,0,11,0)
   CALL PHYSORT(1,0,2,0)
   CALL TITLE(1,PHHARDNESS VS. TIME,100,11,TIME , HRSS,100,9HHARDNESS
   15,10,0,0,7,0)
   CALL XLOG(0,1,2,0,30,0,10,0)
   CALL CURVE(X(1,1),X(1,2*M),9,-1)
   CALL CURVE(X(1,1),X(1,(2*M)+1),9,-1)
   CALL RLNK(10,0,0,0,0,0,7,0,3)
   CALL FNDPL(N)
   N=N+1
   C-
31  C-CONTINUE
   C-CONTINUE
   STOP
103  FORMAT(7F5,2)
   END

```



```

C-
60 10 J=1,N
    READ *.X(J),Y(J)
    IF(1.EQ.1) CALL MARKER(2)
    IF(1.EQ.2) CALL MARKER(1)
    CALL CURVE(X,Y,N,-1)
    N=NC2
C-
65 20 CONTINUE
    CALL MARKER(0)
    CALL CURVE(3.3E1,1.3E-6,1,-1)
    CALL MARKER(2)
    CALL CURVE(3.3E1,1.6E-6,1,-1)
C-
70 C-SFT UP SECONDARY AXES
C-
75 CALL YLGXYS(3.0E-5,2.75,6.75,"MM/CYCLES",-100,5.0,0.25)
    CALL XLGXYS(11.0,5.0,5.15,"MP/A*Z)H*(EH0.5)1/2(EXHY)",-100,-.15
    1.7,0)
C-
80 C-SET UP LEGEND
C-
    CALL MXSALF("GREEK","")
    CALL HEIGHT(0.1)
    CALL MESSAG(LEG5,20,2.4,0.25)
    CALL MESSAG(LEG4,21,2.8,0.55)
    CALL MESSAG(LEG3,30,2.5,0.45)
    CALL MESSAG(LEG2,30,2.5,1.15)
    CALL MESSAG(LEG1,30,2.5,1.45)
    CALL HEIGHT(0.15)
    CALL MESSAG("D*K-STRESS INTENSITY RANGES",100,0.6,-1.0)
C-
90 C-DRAW BOXES AROUND LEGEND AND ENTIRE PLOT
C-
    CALL BLNK1(2,375,5.0,0.0,1.688,1)
    CALL RESET("BLNKS")
    CALL BLNK1(0,0,5.0,0.0,7.0,1)
    CALL RESET("BLNKS")
C-
95 C-TERMINATE PLOTTING
C-
    CALL ENDPL(2)
    CALL DONEPL
    STOP
    END
100

```

APPENDIX I
LSI-11 PERIPHERAL DRIVER PROGRAMS

DSS

```
.TITLE ADAC1
.GLOBAL ADC
.CSECT
R0=R0
R1=R1
R5=R5
PC=27
MOV (R5)+,R0
MOV #10,R1
MOV #1(R5)+,R0
ASH R1,R0
CLR R1
MOV R0,#176770
MOV #176770,R0
INC R1
CMP #20,R1
BNE LOOP
MOV #176772,#1(R5)
BIC #17000,#1(R5)+
MOV R1,#1(R5)+
PC
RTS
.END
```

ADC:

LOOP:

000
J FILE 80AC2
JALCPL DAC
R050
R101
R205
R307
R40V
R50V
R60V
R70V
R80V
R90V
R10V
R11V
R12V
R13V
R14V
R15V
R16V
R17V
R18V
R19V
R20V
R21V
R22V
R23V
R24V
R25V
R26V
R27V
R28V
R29V
R30V
R31V
R32V
R33V
R34V
R35V
R36V
R37V
R38V
R39V
R40V
R41V
R42V
R43V
R44V
R45V
R46V
R47V
R48V
R49V
R50V
R51V
R52V
R53V
R54V
R55V
R56V
R57V
R58V
R59V
R60V
R61V
R62V
R63V
R64V
R65V
R66V
R67V
R68V
R69V
R70V
R71V
R72V
R73V
R74V
R75V
R76V
R77V
R78V
R79V
R80V
R81V
R82V
R83V
R84V
R85V
R86V
R87V
R88V
R89V
R90V
R91V
R92V
R93V
R94V
R95V
R96V
R97V
R98V
R99V
R100V
R101V
R102V
R103V
R104V
R105V
R106V
R107V
R108V
R109V
R110V
R111V
R112V
R113V
R114V
R115V
R116V
R117V
R118V
R119V
R120V
R121V
R122V
R123V
R124V
R125V
R126V
R127V
R128V
R129V
R130V
R131V
R132V
R133V
R134V
R135V
R136V
R137V
R138V
R139V
R140V
R141V
R142V
R143V
R144V
R145V
R146V
R147V
R148V
R149V
R150V
R151V
R152V
R153V
R154V
R155V
R156V
R157V
R158V
R159V
R160V
R161V
R162V
R163V
R164V
R165V
R166V
R167V
R168V
R169V
R170V
R171V
R172V
R173V
R174V
R175V
R176V
R177V
R178V
R179V
R180V
R181V
R182V
R183V
R184V
R185V
R186V
R187V
R188V
R189V
R190V
R191V
R192V
R193V
R194V
R195V
R196V
R197V
R198V
R199V
R200V
R201V
R202V
R203V
R204V
R205V
R206V
R207V
R208V
R209V
R210V
R211V
R212V
R213V
R214V
R215V
R216V
R217V
R218V
R219V
R220V
R221V
R222V
R223V
R224V
R225V
R226V
R227V
R228V
R229V
R230V
R231V
R232V
R233V
R234V
R235V
R236V
R237V
R238V
R239V
R240V
R241V
R242V
R243V
R244V
R245V
R246V
R247V
R248V
R249V
R250V
R251V
R252V
R253V
R254V
R255V
R256V
R257V
R258V
R259V
R260V
R261V
R262V
R263V
R264V
R265V
R266V
R267V
R268V
R269V
R270V
R271V
R272V
R273V
R274V
R275V
R276V
R277V
R278V
R279V
R280V
R281V
R282V
R283V
R284V
R285V
R286V
R287V
R288V
R289V
R290V
R291V
R292V
R293V
R294V
R295V
R296V
R297V
R298V
R299V
R300V
R301V
R302V
R303V
R304V
R305V
R306V
R307V
R308V
R309V
R310V
R311V
R312V
R313V
R314V
R315V
R316V
R317V
R318V
R319V
R320V
R321V
R322V
R323V
R324V
R325V
R326V
R327V
R328V
R329V
R330V
R331V
R332V
R333V
R334V
R335V
R336V
R337V
R338V
R339V
R340V
R341V
R342V
R343V
R344V
R345V
R346V
R347V
R348V
R349V
R350V
R351V
R352V
R353V
R354V
R355V
R356V
R357V
R358V
R359V
R360V
R361V
R362V
R363V
R364V
R365V
R366V
R367V
R368V
R369V
R370V
R371V
R372V
R373V
R374V
R375V
R376V
R377V
R378V
R379V
R380V
R381V
R382V
R383V
R384V
R385V
R386V
R387V
R388V
R389V
R390V
R391V
R392V
R393V
R394V
R395V
R396V
R397V
R398V
R399V
R400V
R401V
R402V
R403V
R404V
R405V
R406V
R407V
R408V
R409V
R410V
R411V
R412V
R413V
R414V
R415V
R416V
R417V
R418V
R419V
R420V
R421V
R422V
R423V
R424V
R425V
R426V
R427V
R428V
R429V
R430V
R431V
R432V
R433V
R434V
R435V
R436V
R437V
R438V
R439V
R440V
R441V
R442V
R443V
R444V
R445V
R446V
R447V
R448V
R449V
R450V
R451V
R452V
R453V
R454V
R455V
R456V
R457V
R458V
R459V
R460V
R461V
R462V
R463V
R464V
R465V
R466V
R467V
R468V
R469V
R470V
R471V
R472V
R473V
R474V
R475V
R476V
R477V
R478V
R479V
R480V
R481V
R482V
R483V
R484V
R485V
R486V
R487V
R488V
R489V
R490V
R491V
R492V
R493V
R494V
R495V
R496V
R497V
R498V
R499V
R500V
R501V
R502V
R503V
R504V
R505V
R506V
R507V
R508V
R509V
R510V
R511V
R512V
R513V
R514V
R515V
R516V
R517V
R518V
R519V
R520V
R521V
R522V
R523V
R524V
R525V
R526V
R527V
R528V
R529V
R530V
R531V
R532V
R533V
R534V
R535V
R536V
R537V
R538V
R539V
R540V
R541V
R542V
R543V
R544V
R545V
R546V
R547V
R548V
R549V
R550V
R551V
R552V
R553V
R554V
R555V
R556V
R557V
R558V
R559V
R560V
R561V
R562V
R563V
R564V
R565V
R566V
R567V
R568V
R569V
R570V
R571V
R572V
R573V
R574V
R575V
R576V
R577V
R578V
R579V
R580V
R581V
R582V
R583V
R584V
R585V
R586V
R587V
R588V
R589V
R590V
R591V
R592V
R593V
R594V
R595V
R596V
R597V
R598V
R599V
R600V
R601V
R602V
R603V
R604V
R605V
R606V
R607V
R608V
R609V
R610V
R611V
R612V
R613V
R614V
R615V
R616V
R617V
R618V
R619V
R620V
R621V
R622V
R623V
R624V
R625V
R626V
R627V
R628V
R629V
R630V
R631V
R632V
R633V
R634V
R635V
R636V
R637V
R638V
R639V
R640V
R641V
R642V
R643V
R644V
R645V
R646V
R647V
R648V
R649V
R650V
R651V
R652V
R653V
R654V
R655V
R656V
R657V
R658V
R659V
R660V
R661V
R662V
R663V
R664V
R665V
R666V
R667V
R668V
R669V
R670V
R671V
R672V
R673V
R674V
R675V
R676V
R677V
R678V
R679V
R680V
R681V
R682V
R683V
R684V
R685V
R686V
R687V
R688V
R689V
R690V
R691V
R692V
R693V
R694V
R695V
R696V
R697V
R698V
R699V
R700V
R701V
R702V
R703V
R704V
R705V
R706V
R707V
R708V
R709V
R710V
R711V
R712V
R713V
R714V
R715V
R716V
R717V
R718V
R719V
R720V
R721V
R722V
R723V
R724V
R725V
R726V
R727V
R728V
R729V
R730V
R731V
R732V
R733V
R734V
R735V
R736V
R737V
R738V
R739V
R740V
R741V
R742V
R743V
R744V
R745V
R746V
R747V
R748V
R749V
R750V
R751V
R752V
R753V
R754V
R755V
R756V
R757V
R758V
R759V
R760V
R761V
R762V
R763V
R764V
R765V
R766V
R767V
R768V
R769V
R770V
R771V
R772V
R773V
R774V
R775V
R776V
R777V
R778V
R779V
R780V
R781V
R782V
R783V
R784V
R785V
R786V
R787V
R788V
R789V
R790V
R791V
R792V
R793V
R794V
R795V
R796V
R797V
R798V
R799V
R800V
R801V
R802V
R803V
R804V
R805V
R806V
R807V
R808V
R809V
R810V
R811V
R812V
R813V
R814V
R815V
R816V
R817V
R818V
R819V
R820V
R821V
R822V
R823V
R824V
R825V
R826V
R827V
R828V
R829V
R830V
R831V
R832V
R833V
R834V
R835V
R836V
R837V
R838V
R839V
R840V
R841V
R842V
R843V
R844V
R845V
R846V
R847V
R848V
R849V
R850V
R851V
R852V
R853V
R854V
R855V
R856V
R857V
R858V
R859V
R860V
R861V
R862V
R863V
R864V
R865V
R866V
R867V
R868V
R869V
R870V
R871V
R872V
R873V
R874V
R875V
R876V
R877V
R878V
R879V
R880V
R881V
R882V
R883V
R884V
R885V
R886V
R887V
R888V
R889V
R890V
R891V
R892V
R893V
R894V
R895V
R896V
R897V
R898V
R899V
R900V
R901V
R902V
R903V
R904V
R905V
R906V
R907V
R908V
R909V
R910V
R911V
R912V
R913V
R914V
R915V
R916V
R917V
R918V
R919V
R920V
R921V
R922V
R923V
R924V
R925V
R926V
R927V
R928V
R929V
R930V
R931V
R932V
R933V
R934V
R935V
R936V
R937V
R938V
R939V
R940V
R941V
R942V
R943V
R944V
R945V
R946V
R947V
R948V
R949V
R950V
R951V
R952V
R953V
R954V
R955V
R956V
R957V
R958V
R959V
R960V
R961V
R962V
R963V
R964V
R965V
R966V
R967V
R968V
R969V
R970V
R971V
R972V
R973V
R974V
R975V
R976V
R977V
R978V
R979V
R980V
R981V
R982V
R983V
R984V
R985V
R986V
R987V
R988V
R989V
R990V
R991V
R992V
R993V
R994V
R995V
R996V
R997V
R998V
R999V
R1000V
R1001V
R1002V
R1003V
R1004V
R1005V
R1006V
R1007V
R1008V
R1009V
R1010V
R1011V
R1012V
R1013V
R1014V
R1015V
R1016V
R1017V
R1018V
R1019V
R1020V
R1021V
R1022V
R1023V
R1024V
R1025V
R1026V
R1027V
R1028V
R1029V
R1030V
R1031V
R1032V
R1033V
R1034V
R1035V
R1036V
R1037V
R1038V
R1039V
R1040V
R1041V
R1042V
R1043V
R1044V
R1045V
R1046V
R1047V
R1048V
R1049V
R1050V
R1051V
R1052V
R1053V
R1054V
R1055V
R1056V
R1057V
R1058V
R1059V
R1060V
R1061V
R1062V
R1063V
R1064V
R1065V
R1066V
R1067V
R1068V
R1069V
R1070V
R1071V
R1072V
R1073V
R1074V
R1075V
R1076V
R1077V
R1078V
R1079V
R1080V
R1081V
R1082V
R1083V
R1084V
R1085V
R1086V
R1087V
R1088V
R1089V
R1090V
R1091V
R1092V
R1093V
R1094V
R1095V
R1096V
R1097V
R1098V
R1099V
R1100V
R1101V
R1102V
R1103V
R1104V
R1105V
R1106V
R1107V
R1108V
R1109V
R1110V
R1111V
R1112V
R1113V
R1114V
R1115V
R1116V
R1117V
R1118V
R1119V
R1120V
R1121V
R1122V
R1123V
R1124V
R1125V
R1126V
R1127V
R1128V
R1129V
R1130V
R1131V
R1132V
R1133V
R1134V
R1135V
R1136V
R1137V
R1138V
R1139V
R1140V
R1141V
R1142V
R1143V
R1144V
R1145V
R1146V
R1147V
R1148V
R1149V
R1150V
R1151V
R1152V
R1153V
R1154V
R1155V
R1156V
R1157V
R1158V
R1159V
R1160V
R1161V
R1162V
R1163V
R1164V
R1165V
R1166V
R1167V
R1168V
R1169V
R1170V
R1171V
R1172V
R1173V
R1174V
R1175V
R1176V
R1177V
R1178V
R1179V
R1180V
R1181V
R1182V
R1183V
R1184V
R1185V
R1186V
R1187V
R1188V
R1189V
R1190V
R1191V
R1192V
R1193V
R1194V
R1195V
R1196V
R1197V
R1198V
R1199V
R1200V
R1201V
R1202V
R1203V
R1204V
R1205V
R1206V
R1207V
R1208V
R1209V
R1210V
R1211V
R1212V
R1213V
R1214V
R1215V
R1216V
R1217V
R1218V
R1219V
R1220V
R1221V
R1222V
R1223V
R1224V
R1225V
R1226V
R1227V
R1228V
R1229V
R1230V
R1231V
R1232V
R1233V
R1234V
R1235V
R1236V
R1237V
R1238V
R1239V
R1240V
R1241V
R1242V
R1243V
R1244V
R1245V
R1246V
R1247V
R1248V
R1249V
R1250V
R1251V
R1252V
R1253V
R1254V
R1255V
R1256V
R1257V
R1258V
R1259V
R1260V
R1261V
R1262V
R1263V
R1264V
R1265V
R1266V
R1267V
R1268V
R1269V
R1270V
R1271V
R1272V
R1273V
R1274V
R1275V
R1276V
R1277V
R1278V
R1279V
R1280V
R1281V
R1282V
R1283V
R1284V
R1285V
R1286V
R1287V
R1288V
R1289V
R1290V
R1291V
R1292V
R1293V
R1294V
R1295V
R1296V
R1297V
R1298V
R1299V
R1300V
R1301V
R1302V
R1303V
R1304V
R1305V
R1306V
R1307V
R1308V
R1309V
R1310V
R1311V
R1312V
R1313V
R1314V
R1315V
R1316V
R1317V
R1318V
R1319V
R1320V
R1321V
R1322V
R1323V
R1324V
R1325V
R1326V
R1327V
R1328V
R1329V
R1330V
R1331V
R1332V
R1333V
R1334V
R1335V
R1336V
R1337V
R1338V
R1339V
R1340V
R1341V
R1342V
R1343V
R1344V
R1345V
R1346V
R1347V
R1348V
R1349V
R1350V
R1351V
R1352V
R1353V
R1354V
R1355V
R1356V
R1357V
R1358V
R1359V
R1360V
R1361V
R1362V
R1363V
R1364V
R1365V
R1366V
R1367V
R1368V
R1369V
R1370V
R1371V
R1372V
R1373V
R1374V
R1375V
R1376V
R1377V
R1378V
R1379V
R1380V
R1381V
R1382V
R1383V
R1384V
R1385V
R1386V
R1387V
R1388V
R1389V
R1390V
R1391V
R1392V
R1393V
R1394V
R1395V
R1396V
R1397V
R1398V
R1399V
R1400V
R1401V
R1402V
R1403V
R1404V
R1405V
R1406V
R1407V
R1408V
R1409V
R1410V
R1411V
R1412V
R1413V
R1414V
R1415V
R1416V
R1417V
R1418V
R1419V
R1420V
R1421V
R1422V
R1423V
R1424V
R1425V
R1426V
R1427V
R1428V
R1429V
R1430V
R1431V
R1432V
R1433V
R1434V
R1435V
R1436V
R1437V
R1438V
R1439V
R1440V
R1441V
R1442V
R1443V
R1444V
R1445V
R1446V
R1447V
R1448V
R1449V
R1450V
R1451V
R1452V
R1453V
R1454V
R1455V
R1456V
R1457V
R1458V
R1459V
R1460V
R1461V
R1462V
R1463V
R1464V
R1465V
R1466V
R1467V
R1468V
R1469V
R1470V
R1471V
R1472V
R1473V
R1474V
R1475V
R1476V
R1477V
R1478V
R1479V
R1480V
R1481V
R1482V
R1483V
R1484V
R1485V
R1486V
R1487V
R1488V
R1489V
R1490V
R1491V
R1492V
R1493V
R1494V
R1495V
R1496V
R1497V
R1498V
R1499V
R1500V
R1501V
R1502V
R1503V
R1504V
R1505V
R1506V
R1507V
R1508V
R1509V
R1510V
R1511V
R1512V
R1513V
R1514V
R1515V
R1516V
R1517V
R1518V
R1519V
R1520V
R1521V
R1522V
R1523V
R1524V
R1525V
R1526V
R1527V
R1528V
R1529V
R1530V
R1531V
R1532V
R1533V
R1534V
R1535V
R1536V
R1537V
R1538V
R1539V
R1540V
R1541V
R1542V
R1543V
R1544V
R1545V
R1546V
R1547V
R1548V
R1549V
R1550V
R1551V
R1552V
R1553V
R1554V
R1555V
R1556V
R1557V
R1558V
R1559V
R1560V
R1561V
R1562V
R1563V
R1564V
R1565V
R1566V
R1567V
R1568V
R1569V
R1570V
R1571V
R1572V
R1573V
R1574V
R1575V
R1576V
R1577V
R1578V
R1579V
R1580V
R1581V
R1582V
R1583V
R1584V
R1585V
R1586V
R1587V
R1588V
R1589V
R1590V
R1591V
R1592V
R1593V
R1594V
R1595V
R1596V
R1597V
R1598V
R1599V
R1600V
R1601V
R1602V
R1603V
R1604V
R1605V
R1606V
R1607V
R1608V
R1609V
R1610V
R1611V
R1612V
R1613V
R1614V
R1615V
R1616V
R1617V
R1618V
R1619V
R1620V
R1621V
R1622V
R1623V
R1624V
R1625V
R1626V
R1627V
R1628V
R1629V
R1630V
R1631V
R1632V
R1633V
R1634V
R1635V
R1636V
R1637V
R1638V
R1639V
R1640V
R1641V
R1642V
R1643V
R1644V
R1645V
R1646V
R1647V
R1648V
R1649V
R1650V
R1651V
R1652V
R16

APPENDIX J
PLM SUPPORT PROGRAMS FOR THE CONTROL LOGIC MICROPROCESSORS

04MAY77 15:51:04 PICARD20

```
640 IF(K(1).LI.48)GO TO 22
650 IF(K(1).GT.57)GO TO 22
660 K(1)=K(1)-48
670 GO TO 21
680 22 IF(K(1).LI.65)GO TO 999
690 IF(K(1).GT.70)GO TO 999
700 K(1)=K(1)-55
710 21 CONTINUE
720 L1=K(1)+16+K(2)
730 L2=K(3)+16+K(4)
740 PRINT 25,(J(1),I=1,8),L1,L2
750 25 FORMAT(8A4,U3,"/",U3)
760 READ(1,2,END=999)MM
770 ENCODE(J,2)MM
780 IF(J(8).NE.N7)GO TO 99
790 GO TO 15
800 99 PRINT 14
810 999 STOP$END
```

READY

[illegible]

```

640  ITEMP(4)=K(1)
650  ITEMP(3)=K(6)
660  ITEMP(6)=K(5)
670  ITEMP(5)=K(4)
680  ITEMP(8)=K(3)
690  ITEMP(7)=K(2)
700  N=K(2)+16+K(3)
710  IF(N.NE.16)IFLG=1
720  ISUM=0
730  NN=N+2+7
740  DO 10 I=9,NN,2
750  ITEMP(I)=K(I+1)
760  ITEMP(I+1)=K(I+2)
770  ISUM=ISUM+ITEMP(I)+16+ITEMP(I+1)
780  10 CONTINUE
790  ITEMP(NN+3)=AND(ISUM,16)
800  ITEMP(NN+2)=AND(ISUM,17)
810  ITEMP(NN+2)=ITEMP(NN+2)/16
820  NN=NN+2
830  DO 11 I=1,NN,2
840  11 CALL STACK(ITEMP(I),ITEMP(I+1),NFLG)
850  IF(IFLG.EQ.0)GO TO 4
860  CALL STACK(11,15,1)
870  I1=I+1
880  I12=I+51
890  DO 12 I=I1,112
900  L(I)=0
910  WRITE("HINFL")L
920  999 STOP:END
930  SUBROUTINE STACK(N1,N2,NFLG)
940  COMMON L(2500),I1,NWC,N27,N18,N9
950  V=N1+16+N2
960  IF(NWC.NE.1)GO TO 1
970  M=SHIFTL(N,27)
980  NNN=OR(NNN,M)
990  GO TO 4
1000  1 IF(NWC.NE.2)GO TO 2
1010  M=SHIFTL(M,18)
1020  NNN=OR(NNN,M)
1030  GO TO 4
1040  2 IF(NWC.NE.3)GO TO 3
1050  M=SHIFTL(M,9)
1060  NNN=OR(NNN,M)
1070  GO TO 4
1080  3 NNN=OR(NNN,M)
1090  4 NNC=NWC+1
1100  IF(NWC.NE.5)GO TO 5
1110  L(I1)=NNN
1120  10 FORMAT(1X,012)
1130  NNN=0
1140  I1=I1+1
1150  NWC=1
1160  5 IF(NFLG.EQ.1)L(I1)=NNN
1170  RETURN:END

```


REFERENCES

1. Crist, J., Drzal, L.T., and Wisnosky, D., "An Automated Liquid Adsorption System for Surface Energetic Measurements", Air Force Materials Laboratory Technical Report AFML-TR-76- , Submitted.
2. Macha, D.E., Sharpe, Jr., W.N., Grandt, Jr., A.F., Submitted to ASTM for presentation and publication at the Ninth National Symposium on Fracture Mechanics, University of Pittsburgh, Pittsburgh, PA, 25-27 August 1975.
3. Tong, H., Bonnett, W.S., Copper, L., Moyer, C.B., Neuner, G.J., "Radiant Heating of Aerospace Materials", Air Force Materials Laboratory Technical Report AFML-TR-73-190.
4. Suchsland, K.E., and Maurer, C.E., "A Unified Ablative Material Thermal Response Analysis Procedure", Air Force Materials Laboratory Technical Report AFML-TR-76-46.
5. Rondeau, R.E., and Ford, D.L., "Use of the APPLE Code in Analyzing the Temperature Response of Aluminum to Nuclear Thermal Radiation", Air Force Materials Laboratory Technical Memo AFML-TM-LP-76-3.
6. Rondeau, R.E., and Ford, D.L., "Coating Analysis with the APPLE Code, Part II: Flight Simulation and Thermal Conductivity Effects", Air Force Materials Laboratory Technical Memo AFML-TM-LP-76-6.
7. Rondeau, R.E., and Ford, D.L., "Thermal and Stress Analyses of the AF Weapons Laboratory Zinc Selenide Laser Window", Air Force Materials Laboratory Technical Memo AFML-TM-LP-76-4.
8. Rondeau, R.E. and Ford, D.L., "Computed Temperature Response of the AFML Flat-Top Laser Window", Air Force Materials Laboratory Technical Memo AFML-TM-LP-76-12.
9. Rondeau, R.E., and Ford, D.L., "Analytical Comparisons of KCl and ZnSe for Laser Window Applications", Air Force Materials Laboratory Technical Memo AFML-TM-LP-76-14.
10. Rondeau, R.E., and Ford, D.L., "Development of AFML Prediction Capability for Laser Effects", AFML-TR-76-243, Submitted.
11. Carlson, R.L., "A 15 Kilowatt CW CO₂ Coaxial Electric Discharge Laser", Air Force Weapons Laboratory Technical Report AFWL-TR-76-115.
12. Kessler, W. C., private communication.

13. Dekaney, A., and Guthrie, L.E., "Determination of the Feasibility of Applying Computer Analysis of Titanium Alloy Microstructures to Quality Control. Air Force Materials Laboratory Technical Report AFML-TR-73-123.
14. Feng, F., Forsen, G., Gillotte, M., "Image Processing Experiments Titanium Alloy Photomicrographs", Pattern Analysis and Recognition Corporation Report No. 76-10.
15. Leet, D.G., "OZ: A Data Acquisition and Analysis System for the Air Force Laser Hardened Materials Evaluation Laboratory Facility", UDRI-TR-77-21, University of Dayton Research Institute, Dayton, OH, (1977).